THE PUBLICATION FOR LIGHTWAVE, INSPIRE, AND AURA ENTHUSIASTS

Sahara: Character Animation, Cloth & Hair

Place with computer magazines

Issue #18 May/June 2000
$9.95 US $14.95 Canada

LightWave® [6]

Allow yourself to be seduced.

Seduced by the power, the elegance and the flexibility. Your first encounter will reveal why LightWave dominates the demanding world of show business.

LightWave has always been the industry standard for exceptional realism. Today it just got better. Much better. Now, with caustics, radiosity and integrated Hypervoxels, LightWave [6] understands and delivers the nuances of lighting and volumetrics better than any other animation system on the market. The addition of a 96-bit floating-point render pipeline assures you ultimate color depth and accuracy.

LightWave [6] introduces the world's fastest and most accurate hybrid inverse/forward kinematics engine. Also, unique to LightWave [6] are IntelligEntities™. These include Endomorphs™ and Skelegons™ used to create and control character animation with efficiency and fluidity.

With the release of LightWave [6] nothing was left to chance. We completely redesigned the program and the process with you, the artist, in mind. The joy of LightWave, all the features, all the functions — better, faster, fun.

The seduction is complete.

NewTek booth #535 at Siggraph '99

www.newtek.com
Call **(800) 862-7837**
Outside the U.S. call (210) 370-8000

LightWave 3D is a registered trademark of NewTek, Inc. Hypervoxels, IntelligEntities, Endomorphs and Skelegons are trademarks of NewTek, Inc. © NewTek, 1999. All rights reserved.

NewTek

ARETE'S
NATUREFX
FOR LIGHTWAVE 3D

ATMOSPHERE | CLOUDS | WATER | WAKES

Deep Blue Sea image courtesy Flash Film Works. ©1999 Warner Bros..

800-544-6599 262-548-8120
www.sharbor.com

800-944-5520 • 702-990-2277
WWW.DYNAMIC-REALITIES.COM

KEYFRAME MAGAZINE

Editor
Doug Kelly *dakelly@earthlink.net*

Advertising Rep
Tiffani Banaszak *keyframe1@aol.com*

Co-Conspirators, Writers, and Artists
Tiffani Banaszak *keyframe1@aol.com*
Ron Dutcher *rd@mbs-clinic.com*
Graham McKenna
graham@grahamprismatic.free-online.co.uk
Brenden Mecleary *bman2@ix.netcom.com*
Sean Moyer *seanmoyer@mediaone.net*
Steve Pugh *steve@foundation-i.com*
Lee Stranahan *lee@stranahan.com*
David Warner *dwarner@albany.net*
Marc Williams *marc@spydaman.com*

Proofreader
Patty Kelly

The Digital Underground
DMG Publishing
2756 N. Green Valley Pkwy. #261
Henderson, NV 89014
702-990-8656

How To Get It
888-778-WAVE
http://www.keyframemag.com

Keyframe Magazine is published 6 times annually by DMG Publishing. © 2000 DMG Publishing. All rights reserved. Reproduction, in part or in whole, without written permission is strictly prohibited. Printed in the USA.

The Secret Formula
Keyframe Magazine is created using Quark 4.1, Photoshop 5.0, LightWave 6.0, and various plug-ins.

No dongles were harmed in the making of this publication.

Departments

3 Editorial — by Doug Kelly

4 Ear to the Wall
- Foundation Institute
- NewTek at NAB
- VideoToasterNT.com
- Graphic Board Certification Program
- Mac OS X
- QuickTime

6 On Topic — by David Warner
- L6.0b, Lighting, BRDF Shader and more

26 Reviews
- IFW Shaders
- Cinematte

32 Final Cut — by Steve Pugh
- The Men Behind the Curtains

Tutorials

8 Programmer's Hideaway — by Brenden Mecleary
- LightWave 6.0 Compatibility

10 Tail to be Told — by Graham McKenna
- Simulating fur with textures

13 All That Glitters — by Ron Dutcher
- Making gemstones

16 Sahara — by Tareq
- Cover Case Study

22 Web Animation — by Marc Williams
- Rendering animated buttons

23 Skeleton Modeling — by Sean Moyer
- Complex objects from basic tools

EDITORIAL

New Tools, New Habits

It's been a relatively slow month for news and gossip about LightWave, Aura and the Video Toaster. My email isn't piling up as fast as it usually does, the traffic on Flay.com and the LWML is down slightly, and there aren't as many flame wars on c.g.a.lightwave. I think this is mostly due to everyone's heads being down while they learn how to use the new tools. There are a lot of new toys in the pipeline these days, and figuring out where they fit (or don't) in your personal workflow can take some time.

Most of us are finding that formerly trustworthy old tools are now either broken or awkward. Brenden Mecleary explains some reasons for broken software in this issue's Programmer's Hideaway. NewTek is making progress on fixing problems, having released the 6.0b update and having started a Graphics Board Certification program. Addressing the awkwardness issue is often more a matter of retraining yourself than of changing your tools. Dave Warner's On Topic column in this issue includes a few examples of old habits that die hard, and how to put them out of your misery. If you look at the new tools with an open mind and a problem-solving attitude, I think you'll be pleasantly surprised at how much easier your work can become.

The usual group of early adopters has, of course, been producing some amazing art with the new tools. This issue's cover shows what a talented artist can do with the integration of LightWave, Messiah, and Shave and a Haircut. The learning curve on these applications is no joke, but the results can be well worth the effort.

Putting in a little extra effort right now may pay off handsomely in the not-too-distant future. The last year has seen a significant advance in cinematography technology. The new High Definition (HD) cameras, particularly those designed to shoot the 24 frame per second progressive (24p) format, will change the way many films are made. More than one 'film' has already been shot on HD, edited and post-produced using video tools, then converted to film for distribution.

The implications for the effects industry are staggering: for the rental price of a top-end HD 24p camera, you can have film-quality digital footage ready to feed directly into your workstation, without the enormous up-front costs of film scanning or the quality loss of telecine. This means that anyone who can scare up a moderate-to-high video production budget can shoot on a format that can be upgraded to film quality *after* a distribution deal is signed. This is an independent filmmaker's dream come true.

More importantly for the LightWave community, this means that film-quality special effects are going to be easier, cheaper and faster to do, simply because the film scanning/telecine process isn't bogging down the front end. There is going to be more demand for f/x that will stand up to HD and film presentation. That means the artists who limit their performance to the standard video level won't make the cut, and those who perform at the HD and film level will find themselves even more in demand.

Where do *you* want to be when the dust settles?

Ear to the Wall

News from the LightWave arena

Foundation Institute

Foundation Imaging is developing a full service educational center located at its production facility in Valencia, California to teach the Foundation 3D Exchange Curriculum. In cooperation with 3D Exchange, the world-renowned computer animation training school, Foundation Institute will offer a multi-tiered educational program including beginner through expert classes from one week through a two-year full certification program. The curriculum will be based upon 3D Exchange's standardized, industry recognized program emphasizing NewTek's LightWave 3D software package.

The Foundation Institute was formed to provide the next generation of LightWave training. "By combining the superior training methods developed at 3D Exchange in Alameda, Ca. with the incredible level of professional artistry Foundation Imaging is known for, we can offer our students an ideal learning situation," says program director Karen Tosoni. Foundation General Manager Steve Pugh adds, "We are very excited about this partnership. Foundation Imaging has been active in animation education through our trade publication articles, internships, and LightWave training tapes. This school will be taking that idea to the next step".

The updated 3D Exchange curriculum incorporating Foundation Imaging guidelines will be available for purchase through a NewTek Authorized Training Center only. The new version will be available in Summer 2000.

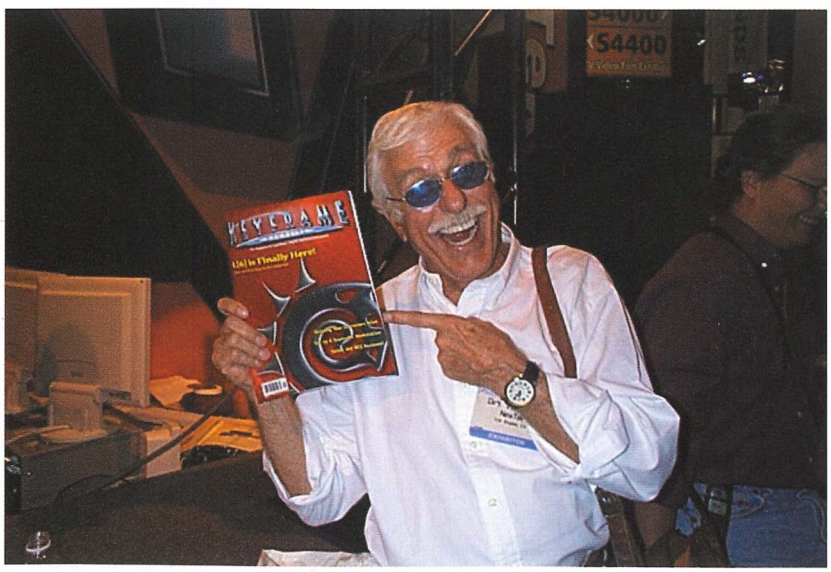
For further information please contact:

Karen Tosoni
Foundation Institute
24933 West Avenue Stanford
Valencia, CA 91355
661-257-0292
www.foundation-i.com

NewTek at NAB

Those who attended NewTek's Technology Preview before NAB 2000 were the first to see the new Video Toaster NT 2.0, Aura 2.0 and more. According to Paul Lara, Video Toaster Product Manager and *Toaster Boaster Extraordinare*, "There are some exciting developments ahead for the Toaster community, and some exciting announcements for the entire NewTek family as well!"

NewTek has also just started a Graphic Board Certification program. NewTek is working closely with leading graphics cards vendors (including 3dfx, 3DLabs, ATI, ELSA, Evans and Sutherland, Intense 3D, Matrox and Nvidia) to make sure that everything from lens flares to the various display and texture modes look great, and to help design specialized drivers that take full advantage of LightWave [6]. Upon acceptance to the program, vendors will receive a LightWave certification with a special LightWave [6] certified logo for use in their packaging as well as collateral materials.



Issue #17 was very popular at NAB 2000.

Safe Harbor Unveils New VideoToasterNT.com Website

Safe Harbor Computers, Inc. is proud to present the redesigned VideoToasterNT.com website, an innovative web site to provide information about the hottest NewTek products. VideoToasterNT.com was developed by Safe Harbor to promote a technological community made possible by the Internet and to allow for current and potential users to come together and share tips and information, and to get expert, accurate advice about this product.

The new Video Toaster NT web site (http://www.videotoasternt.com) was designed to provide up-to-date, accurate information about the Video Toaster and to provide detailed product specifications. The site's interactive forum allows for questions to be answered by NewTek and Safe Harbor representatives as well as other Video Toaster NT users -- and now it's better than ever!

What's New

* New look and feel
* Completely redesigned "What is a Video Toaster?" section. Each component is broken down into readable, digestible pieces. Navigation is quick and easy. Even the novice can learn all about what the Video Toaster has to offer!
* Streaming videos of Don Ballance demonstrating the Video Toaster NT 1.0
* New answers to frequently-asked questions

As always, available at the site are technical support documents for the Video Toaster NT, and tips for getting your Video Toaster NT system up and running. In addition, this site offers instant system quotes on custom turn-key complete Video Toaster NT systems from Safe Harbor and easy click ordering. The Video Toaster NT can't be made any easier!

About Safe Harbor Computers

Established in Waukesha, Wisconsin in 1987, Safe Harbor Computers is home to graphics and video products for Windows, Mac, and DEC Alpha based computers. With friendly, knowledgeable sales and technical staff, Safe Harbor continues to set the standard in digital video sales and service. Visit Safe Harbor online at http://www.sharbor.com or call 1-800-544-6599 for more information about products and services.

Welcome to Video Toaster NT .com - Microsoft Internet Explorer
A MEMBER OF SAFE HARBOR'S FAMILY OF WEB SITES!
What is a Video Toaster NT?
Frequently Asked Questions
Discussion Forums
Plugins & Add-Ons
Technical Support
Toaster Links
Design Your Toaster NT System
Preorder Toaster NT 2.0
The original, unofficial Video Toaster NT enthusiasts community!
Video Toaster is a registered trademark of NewTek, Inc.
http://dvguide.sharbor.com/toasternt/support/toaster103.html
Trusted sites

Apple News

Developers came to Apple's Worldwide Developer's Conference to learn as much as they could about Mac OS X (now slated for a public beta this summer). Committed to a gentle migration from Mac OS 9, Apple pledged to deliver a super-modern OS to customers, one that is simple, elegant, and extremely stable and reliable. It will offer "killer graphics," designed both to delight customers and to encourage developers to innovate and astound. And it will provide customers the best Internet experience possible, including support for Java 2.

Over 200 developers have announced support for Mac OS X. (www.apple.com/macosx)

During his keynote, Steve Jobs announced that "Nielsen//NetRatings has named Apple's website, www.apple.com, the most popular destination for movie trailers on the web." That's testimony both to the quality and popularity of QuickTime. There are now well over 50 million copies of QuickTime on Macintosh and Windows desktops around the world; and this summer, Apple expects to release a new version of QuickTime that will make web animation, movie watching, music, and VR panoramics even more enjoyable for watching and listening.

And as for WebObjects, Apple announced aggressive new pricing for its application server software, reducing its price from $50,000 to $699.

L6.0b, Lighting, BRDF Shader, Fading Surfaces

by David Warner

It's no surprise that LightWave 6.0 continues to be the hottest topic in the LightWave community. With the release of the first major update, version 6.0b, L6 is looking better than ever. Numerous bug-fixes, improved stability, major speed increases, and an even longer feature list are just a few of the things L6b users are enjoying.

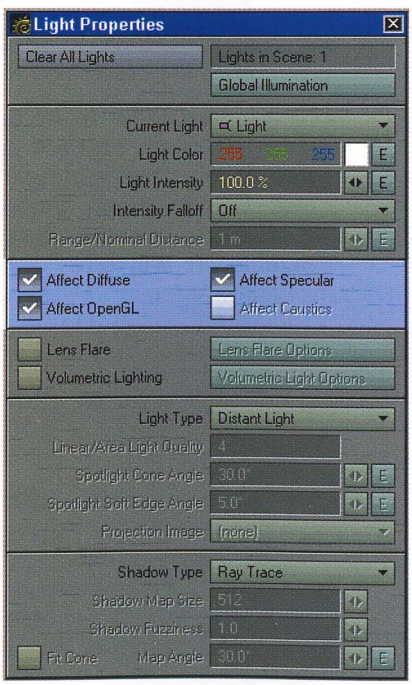
The LightWave 6.0 version B update has arrived!

Kudos to the NewTek programmers for ignoring the noise that many people on the Internet were making about the L6 update being "late." It's a shame that some people are so quick to attack NewTek. We all know that the LightWave programmers are busting their butts to get these software patches done and out the door, and it just doesn't make sense to rush these things and release buggy or untested code. I guess the fast-paced lifestyle of the Internet has made many people remarkably impatient. However, NewTek did the right thing and did not release version 6.0b until it was ready.

While the core components of L6 are shaping up very nicely, there are still many rough areas for LightWave add-ons. Unfortunately, there are a large number of plug-ins designed for LightWave v5.6 that do not work correctly with L6. These are known as 'legacy' plug-ins, and the task of getting them to work is mostly the responsibility of the creator of each plug-in.

By now, any plug-in manufacturer that is still supporting its product is most likely aware of any L6 compatibility problems and (I hope) already working on an update. However, there are many cool plug-ins out there that were released as freeware by LightWave users who just like to program and give away cool stuff to the LightWave community. I'm sad to say we may never see all of these plug-ins converted over to the L6 environment, mainly because most of these generous people have moved on and simply don't have the time or interest to rewrite their plug-ins. We can consider these to be some of the growing pains of the great L6 migration. In a few months or years we will have forgotten all about the painful transition to L6, because the improvements and advances are well worth it!

With this issue, I will begin to focus more on LightWave 6.0 questions. To those of you who are not using L6 yet, I can only offer strong encouragement to make the upgrade. Once you get used to the many new changes in L6, you will never look back!



The L6 Lights panel with the Affect Diffuse and Affect Specular options highlighted.

Q I need to fine-tune the lighting in my scene and to adjust each light without seeing the effects of other lights in my scene. Is there a way to turn off each light without having to delete it from my scene?

A One way is simply to lower the light intensity down to 0%. However, if you have envelopes applied to each light, or if you do not want to change intensity values that took a lot of experimentation to get correct, then this will not work. A much better solution is just to un-check the Affect Diffuse and Affect Specular options, which are visible in the Lights panel for each light in your scene. Turning off these options essentially has the same effect as lowering the light's intensity down to 0% or turning the light off.

Q I have an object with dozens of different surfaces and I need to make each surface become transparent and fade away over time. What's the easiest way to do this?

A In LightWave 6.0 you have the ability to apply an envelope to any of the surface attributes, so it is very easy to make individual surface values (including transparency) change over time.

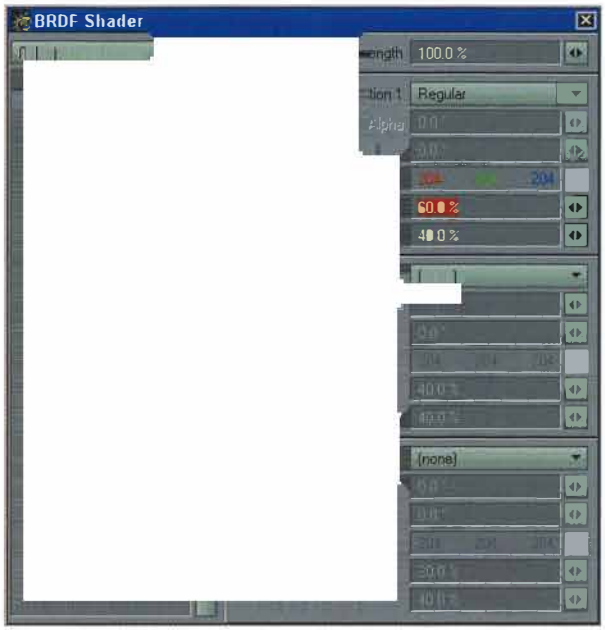
However, if you do not have L6 yet, then probably the best way to do this would be to take all of the polygons with different surface names and turn them into individual objects, so if you had a dozen different surfaces in your object, you would end up with a dozen different objects. Once the surfaces are separated into different objects, you can then use the Dissolve envelope in the Objects panel to dissolve out the different surface objects over time.

You can easily separate the surfaces into different objects by selecting all the polygons with a specific surface name, cut and paste them into a new layer in Modeler, and save that layer as a new object. Repeat this for each of the surfaces in the object. If you have access to the Web, the good folks at Station X Studios have made available a free plug-in that automates this entire process. Just go to www.stationxstudios.com and you'll find the plug-in in the Store->Software->Free Stuff section.

Q Is there a way to prevent an object from being affected by a particular light? I'm using L6 but I have not been able to figure out any way to do this.

A Unfortunately, even with LightWave 6.0, there is no easy way of hiding an object from a light in your scene. Probably the easiest way to accomplish this effect is to simply render the animation in two separate passes. One pass would have the object by itself with the light turned off, and the other pass would have all the remaining objects in

the scene with all the lights turned on. This will accomplish exactly what you are attempting but it's a cumbersome solution, at best.



The BRDF shader interface.

L6 does have a new surface shader that will allow you to turn off the influence of individual lights. If you go to the Shaders sub-tab in the Surface Editor panel, then click on the Add Shader pull-down menu, you will see a shader called BRDF. Add this shader to each surface in the object that you want to have unaffected by the light, then double-click on the shader's name to access the options area for that shader. This is a very powerful shader plug-in which can do much more than hide surfaces from a light, but it will do exactly what you are attempting. The only downside to using this shader is that if you have an object with a lot of different surfaces, then it will take more mouse clicks to add and edit this shader for each individual surface.

Q I am very unhappy with the way Smooth Shift works in L6. In older versions, all I had to do was select Smooth Shift and a dialog box would appear, allowing me to enter the amount I want to shift polygons. Now I have to go through many extra steps to get the same result. What am I missing here?

A You're missing a *lot*! Many of the tools in Modeler are now fully interactive, allowing you to make changes to the object on-the-fly, without having to type numbers into a dialog box. For example, when you select the Smooth Shift tool in L6, just hold down the right mouse button and move the mouse to interactively adjust the shift amount. This same concept holds true for many of the functions in Modeler. Once you get used to this different way of doing things, you won't ever want to go back.

Q I'm trying to render out a very low resolution AVI animation, but LightWave keeps giving me the error message "Image format not supported by the selected AVI compressor". Is this a problem with L6 or something else?

A No, it is not a problem with L6, but rather a limitation of certain AVI codecs. Many compression codecs, especially the older ones, can only save animations with a specific aspect ratio or a width-to-height ratio. So if you were trying to render out an animation that was 200 pixels wide by 200 pixels high, you might find that many of the AVI codecs will cause LightWave to return an error message.

There really isn't a solution to this problem other than to experiment with the different AVI codecs to find one that will save at the resolution you are working in. Typically, most of the newest codecs, like any of the recent Intel Indeo versions, will work with any sort of animation you are generating. The trick is to find a codec that will work smoothly on the target playback machine. If the animation is only for your own personal use, then you can use whatever codec works best on your machine. If you are generating something that will play back on other machines, such as a commercial CD-ROM, then you need to use a compression codec that everyone has and that you are certain will work reasonably well on the target playback systems. This is not an easy feat to pull off, and I can only recommend that you search the Internet for advice on which compression codec is best for your project.

Q I do not see the MetaForm Shaper and Target plug-ins in L6; where are they?

A These plug-ins are no longer needed because Layout is now able to directly load MetaNURBed objects. You are able to adjust the weight of the object's mesh on-the-fly, which means that you can set a very low weight for low-res preview renders and then increase the weight, so that the object looks very smooth and nicely detailed for the final render.

Q If I change any of my video card settings, such as resolution or color depth, while L6 is running, the interface does some really crazy things. LightWave does not crash, but the program becomes pretty much unusable. Is there a way to fix this?

A This problem only seems to occur with video cards that allow you to change your display settings without rebooting, so for most people this problem will never pop up. At this time the only sure-fire way to prevent this from happening is to not change your display settings while LightWave is running. This is one of the major drawbacks of having such a non-standard interface on LightWave. I hope the programmers will someday see the value of redesigning the interface so that it plays nicely with the standard Windows and MacOS display settings.

Q I'm trying to use the new L6 version of the SaveMorf displacement plug-in to save the changing shape of an object that is being affected by bones. The plug-in appears to work okay, saving out the correct number of objects, but when I try to load any of these newly saved objects, I get an error message that LightWave is unable to read the object files. What am I doing wrong?

The SaveMorf plug-in interface.

A Usually the Morph Saver plug-in is only used to export objects that are created with the new Endomorph tools in L6. However, if you set things up correctly, it can indeed be used to save any changes to an object's shape or even its position. The trick is to set up a single Endomorph target without any changes in the object's shape, then define a single keyframe in the Morph Mixer plug-in to set up a morph to the Endomorph target. Even though the object's shape is not affected by the Endomorph envelope, the Morph Saver plug-in will respond correctly now; and when you enter a new filename to save the object sequence as, everything will work exactly the way you want it to.

For those of you who enjoy reading this column and find a great benefit from the knowledge exchanged on the LightWave Mailing List, I encourage you to check out a new product my company has just released:

Continued on page 30

LightWave 6.0 Compatibility

by Brenden Mecleary

"Backward compatibility, n. 1. Not crashing. See also: 'Shippable'"
—from the NewTek Abridged Dictionary of Software Development

Not long after I sent the last installment of this column off to Keyframe, Dynamic Realities gave me a shout and asked about making my plug-ins compatible with L6. As some of you may know, I wrote the RealiTools and RealiTools 2 packages that DR sells, as well as the LumeTools package (which was a port of the Softimage plug-ins).

So, L6 hit, and stuff broke. As I mentioned in the last column, I don't have L6, so DR hooked me up with a temporary license and off I went. Suffice it to say I was less than pleased with L6. But I'll let the dedicated reviewers tear it apart for me. However I felt about dealing with L6, I still had work to do.

"RealiTools - crash on everything" Not very promising. "Crash on everything" were the exact words used to describe to me the state of RealiTools in L6. As I suspected when I first read that, it turned out to be the result of bad version checking on my part. As I've mentioned before in this column, *never allow your plug-in to activate with a version number you don't support.* It sounds obvious now, but three years ago I didn't understand the implications of doing otherwise. So if your plug-in wants API version 2, return `AFUNC_BADVERSION` for *any* other version number. In other words, use `if(version != 2)` instead of `if(version < 2)`.

This was a serious problem in RealiTools, which generally supported up through API version 3, but failed to return `AFUNC_BADVERSION` when called with version numbers higher than that. *Mea culpa.* Fortunately, RT2 and Lume were better about this. Once I fixed the version checking in RT, most of the crashes went away, and the rest of the problems (I don't recall any other GPF-level bugs) were relatively easy to fix.

"What is the sound of -1 hand clapping?" As you probably know, L6 has a new render pipeline that uses floating-point numbers for each pixel's RGB values. A side effect of this is that L6 allows you to have pixels that are brighter than 1.0 (or integer 255), and darker than 0.0 (or integer 0). Accordingly, the `BufferValue` type in the LW SDK changed from `unsigned char` to `float`. This is just fine for plug-ins built for 6.0. But plug-ins that have to run on both 5.6 and 6.0 must continue to use `unsigned char`, so that they remain compatible with old versions of LightWave.

So what exactly *is* the value 256 as an `unsigned char`? Depends who you ask, and in which context you ask. If you ask me, in the context of a rendering pipeline, 256 as an `unsigned char` is 255. So is 257. And 258. Likewise, in the same context, -1 as an `unsigned char` is 0, as is -2, -3, etc. But compilers don't know about context, and they wouldn't really give a damn if they did. So, the compiler says 256 as an `unsigned char` is 0, 257 is 1, 258 is 2, etc. On the other side, -1 is 255, -2 is 254, -3 is 253, and so on.

Why do I bring this up? Because L6 follows the rules of the compiler. For old plug-ins, the returned `BufferValue` is simply cast from its floating point value (times 255) to `unsigned char`, *regardless of whether the result will actually fit!* This means that if L6 renders a pixel that is brighter than 1.0 or darker than 0.0, your plug-in *will* handle that pixel incorrectly. And the plug-in has absolutely no way of knowing what that 0xE7 value it just got really means - negative 25, or positive 231. This leaves us with something of a dilemma. Do we:

A) Re-write all image filters and pixel filters to support 6.0? If we do this, we duplicate all our filter code - remember, we still have to support 5.6 - and risk breaking code in our own plug-ins that can't handle out-of-bounds values. It also takes a lot longer.

B) Re-write all image filters and pixel filters to support 6.0 *only enough* to get the floating-point pixels and clamp them from 0 to 255 before handing them to the old code? This still risks breaking code, and takes a significant amount of time to retrofit multiple plug-ins. It also loses the overcranked pixels. (Though, to be honest, I'm still not convinced that very many people *need* overcranked pixels.)

C) Write a new image filter plug-in that users have to apply by hand before any old image / pixel filters? This would serve the same function as described in (B), and clamp pixels from 0 to 255. It's ever so slightly more inconvenient for the user and still lops off overcranked pixels, but it can be written much faster, and doesn't involve changing the code for the old plug-ins, thus reducing risk of breakage.

D) Tell customers, "tough tamales, blame NewTek"? About the only way this would be productive is if hordes of customers realize that yes, in fact, this *is* NewTek's fault, and harass them mercilessly until they fix it. And if you honestly think either of those things will happen, I have some land in Florida that might interest you.

DR and I decided to go with option C. This means that we can get the patched plug-ins out more quickly, take less risk of breaking them and having to issue *another* patch, and not do a whole bunch of work for an uncertain amount of return.

"Yin and Yang" Open and close. Up and down. Init and Cleanup. Hmm. LightWave has always had problems calling plug-ins' `Init()` and `Cleanup()` functions properly, particularly in Screamernet. In L6, it's totally bugged beyond hope. This becomes alarmingly obvious with surface shaders, whose `Init()` and `Cleanup()` functions also get called when L6 renders the surface preview sphere.

The lesson is this: *your `Init()` and `Cleanup()` functions can and*

will be called at any time, in any order, and sometimes not even called at all. Accept this and code accordingly. I was forced to go back and fix some code as a result of this, but the result of the fixes was the way I should have written the code in the first place.

Panels Issues Most of the fixes I ended up making for L6 were panels fixes. I'll try to go through them relatively quickly. By the time I got around to fixing up Lume, I finally broke down and made some macros to detect whether LightWave was running LWPanels for 5.6 or 6.0. Trying to write code that works for both just isn't always possible.

```
#define LUME_PANELS_VERSION_IS_LW6(panl, panID)                \
        (PAN_GETVERSION(panl, panID) == 18)
#define LUME_PANELS_VERSION_IS_LW56(panl, panID)    \
        (PAN_GETVERSION(panl, panID) < 18)
```

One thing DR did in RealiTools 2 and Lume was add "About" and "Help" buttons to the command bar at the bottom of each panel. For some reason, this didn't seem to work in L6. It seems that LWPanels won't let you move a button into that area any more. And, because the "OK" and "Cancel" buttons are now centered at 1/4 and 3/4 the width of the panel, the X position of the buttons wouldn't work either. So for L6, I just ended up positioning those buttons in the lower-right corner of each panel. Which brings me to the next bug...

Panels seems absolutely clueless about the correct dimensions of a panel before it is opened. (I'm forced to wonder if it even gets the correct dimensions *after* it's opened.) This made positioning controls in the lower right very difficult, because the panel invariably reported its size as much smaller than it actually was. The only solution that I could think of for this was to go through *every* control in the panel, get its X and Y extents, then use those as the approximate panel width. It seems to have worked...for the moment.

`MINIRGB_CTL`s used to incorrectly report their width as being *much* greater than they actually were. Somewhere on the order of 30 pixels. To position an envelope button next to a `MINIRGB_CTL`, you would have to get the right edge of the control (as reported by panels) and subtract 30. This was fixed in L6, causing all the carefully positioned envelope buttons to get placed somewhere *inside* the boundaries of the `MINIRGB_CTL`. So don't subtract away that space any more if you're running under L6.

Control spacing has been decreased. In LightWave 5.6, you could move controls up 8 pixels on the Y-axis if you wanted them stacked closely. In 6.0, you should only move them up by 4 pixels.

`LIST_CTL`s have changed their behavior for calling their callback functions. It is common to have some user data tied to your list control, so you can tell it how many items it contains, and what the text is for any given item. In L6, the `LIST_CTL` will call your callback functions *before you get a chance to set your user data pointer!* So make sure your callbacks can tolerate a `NULL` user data pointer.

The labels on `AREA_CTL`s have disappeared into the wild gray yonder.

"Does this hurt? What about this?" `LWImageID`s don't seem to behave like they used to. LightWave 5.6 would obligingly open multiple copies of the same image, and return unique `LWImageID`s for each of them. L6 does not seem to do this. Instead, it only loads one copy of an image. I didn't spend much time investigating this, but I'm not sure L6 does any reference counting, either. Could be bad.

Another behavior of the image list has changed, as well. In prior versions of LightWave, the `LWImageList->load()` function would return an absolutely useless `LWImageID`. I've heard that you can just add one to get it to work, but if *that* is fixed, then your code is broken again. So I never touched the thing. Instead, I would search the image list for the last image, load the image I wanted, then seek to the next image. This worked fine in LW5.6, because you could call `LWImageList->next()` with an `LWImageID` of `NULL` to get the first image in the list. L6 no longer supports this behavior. I finally settled on re-writing my image load function to search for the last image with the same file name. It seems to have worked...for the moment.

The last bug I ran across - a rather strange one, I might add - deals with image filters trying to retrieve buffers they didn't specify in their `flags()` function. For some reason (don't ask me why; I plead temporary insanity), one of the Lume image filters would *always* retrieve the LightWave special buffer, even if it didn't intend to use it. The filter *does* return the correct *flags* - if it doesn't intend to use the special buffer, it won't request it from the `flags()` function. LW5.6 seemed to have no problem with this, and presumably it returned either NULL or a garbage pointer when the buffer was requested. Since the pointer was never used, this didn't cause any trouble.

L6 broke this, however. When the plug-in tried to request the special buffer (but didn't specify it in the `flags()` function), L6 crashed. I don't know why. Yes, returning `NULL` *would* have been a more rational response on its part. Oh well. I fixed the code so that it doesn't request the special buffer if it's not going to use it. I don't know who to be mad at on this one. It was stupid of me to request a buffer I wasn't going to use, but it seems just as stupid to crash as a result of said request.

"Memories...pressed between the pages of my source..." Revisiting all this old code has been an odd experience for me. RealiTools, in particular, has been quite an eye-opener. Impress your friends at SIGGRAPH with this little bit of trivia: which plug-in package was implemented almost entirely in header (.hpp) files? RealiTools. (Can you say "newb"? I thought you could.)

Another one: which surface shader plug-in allocates memory *at every single pixel* on the surfaces to which it's applied? That would be RealiTools reflective blur plug-in. What amazes me most about that one is that it doesn't seem to have drastically affected the performance of the shader. The allocations were all relatively small, on the order of a few hundred bytes, so perhaps the allocator can deal with those reasonably efficiently. Or perhaps the overhead of allocating memory is just small relative to the overhead of tracing several dozen rays per pixel. Yes, it's fixed now.

I'm not sure if I'll get around to updating the free plug-ins on my web page any time soon, if ever. I'm already burnt out enough just having to hassle with L6 to fix the RT/RT2/Lume sets. It's been painful. Maybe later.

All said, while it's always somewhat embarrassing to go back and confront old code, it is also comforting to realize that I have in fact learned something in the last three years. Like what .cpp files are for.

Brenden Mecleary finally got back into the games biz. Now he gets to write stuff for the Playstation 2. E-mail questions, comments, and flames to bman2@ix.netcom.com.

A Tail to be Told

by Graham McKenna

Too often we see exceptional modeling of human characters and fantastical creatures only to find our eye telling our brain that there is something not quite right with what we are viewing. A lot of this is due to lack of photorealism in certain surfaces that in the past has proved to be most elusive, namely fur and hair.

Usually to achieve such surfaces we have to lean towards third party plug-ins like Metrografx' Fiber Factory or Dynamic Realities' Particle Storm, both of which work with two-point-polygon strands. This can amount to certain problems due to LightWave's calculations on two-point polygons. First of all, the nature of hair, with its usual entanglement, does not lend itself to speed where the renderer is concerned. LightWave treats these strands in the same way it treats particle one-point polygons, meaning that if any particles overlap, it causes the calculations to skyrocket, especially if they have any transparency applied. With two-point polygon strands, the problem is multiplied as there is more chance of overlapping due to the strands having length. Additionally, their size is based on distance from the camera. This can cause huge perspective problems if the object passes by the camera in to the distance.

Another downfall to this approach is the lack of ability the renderer has in raytracing one or two-point polygons, meaning it won't show up on any reflective or refractive surfaces. Meanwhile, it's also unable to trace the shadows (both cast and self shadowing). Even after all these drawbacks, the strand object still does not look soft. For it to achieve such a look, there is usually another branch to the plug-in in the form of a pixel filter which applies itself at the end of each antialiasing pass. This lets the plug-in pass information from the strands for shape, while using the pixel filter for hair attributes, which in turn are passed to the renderer. This usually equates to large rendering times due to the fact it has to do all the calculations just to display the strands on each rendering pass, while also needing further calculations for the pixel filter on each anti-aliasing pass.

With the rate at which software revisions are released, there are sure to be better solutions in the future. Already there are Shave And A Haircut by Joe Alter and soon-to-be-released Sasquatch from Worley Laboratories. However, the point of this exercise is that in most cases the ability to soften the edges of your objects to create hair, fur or fuzz is already there within the base interface supplied by NewTek. So put your wallet back in your pocket while I take you through the procedure step by step. The following steps will result in our very own furry tail.

Step 1

First what we will need is our geometry for our effect to work. By clicking the ball option while in modeler we are confronted with various options for creation of our sphere. Using the default sizing for our sphere is a good basis to start from. The last state of this tool will determine the sphere type, namely globe or tessellated sphere. The globe option would result in the mesh being created using quads to build the geometry. Although it is clean in appearance, we will have to keep in mind that farther down the line in the proceedings we will have to deform the geometry through the use of displacements maps which would result in non-coplanar rendering errors due to the geometry's lack of triangles. For this reason, we do best to opt for the tessellated sphere, which by its nature originates triangulated, and this will eliminate any such errors further into the technique.

For the purpose of this effect, we will create the sphere at a tessellation level of two. Higher tessellation levels would result in smoother deformation when we apply the displacement map (which isn't necessary for this particular effect but does have its uses). Now taper and stretch the sphere to resemble the geometric shape illustrated in Figure 1. You will also notice in Figure 1 a second shape which is smaller than our initial shape. The purpose this serves will be to morph our initial shape into the secondary shape, which will be covered in more detail further into the technique. For our needs, we can use our initial shape and downsize it by 50% creating our morph target in one easy step.

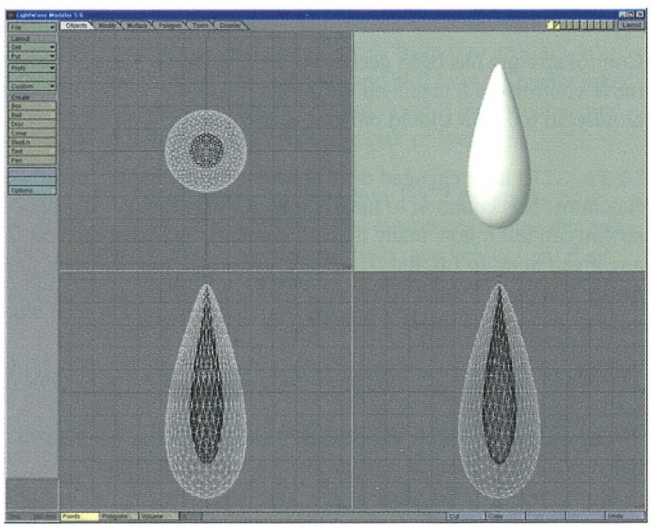


Figure 1 Tessellated sphere and morph target

Step 2

Now that we have our geometry, we should load each in turn into layout,

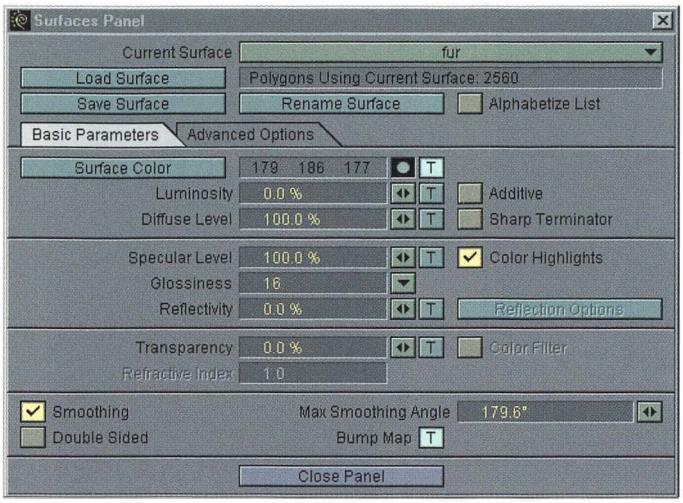


Figure 2 Surface attributes for our furball.

loading our base object first followed by the morph target. While in the objects panel, there are a couple of things we should do, namely turning off any shadowing and ray tracing options for our morph target. This in effect disables it from the renderer. Once we have disabled these options, we should exit the objects panel and set up our base surface for the effect. Figure 2 shows our surface panel attributes for our furball.

At this point, we are looking for certain attributes to be in place, namely low glossiness value for a soft sheen to our surface and a roughness which resembles stone. To achieve the rough texture we will apply a couple of procedurals. The first of these is fractal bumps, which is applied at very fine sizing (around 10 to 20 mm) which gives the surface a very grainy appearance. The second surface to apply is crumple, to give large bump veins across the surface. Texture sizes around 40 cm will give good results for what we need. Both these textures are applied with a large percentage of bump amplitude; above 500% isn't out of the ordinary, but varying amplitudes give varying effects and are worth trying. What to keep in mind when using such high amplitudes is that it is going to effect the speed of the render. For this reason, it is advisable to start applying the texture at smaller amplitudes and to increase these until you get the look you are happy with. This should give a good gauge of what amplitude is actually needed for the look; there is no sense in bumping up the amplitude straight off if the same look could be achieved at half the amplitude. Our specular values should be low in glossiness with color highlights switched on; around 16 glossiness value will give a soft specular highlight. We also add a cylindrical image color map (illustrated in Figure 3) to the surface to break up its color appearance, adding to the effect.

Figure 3 Color image texture

Figure 4 shows a render of our partially finished surfacing which will be fine tuned as we progress to enhance the effect.

Figure 4 Partially surfaced shape

Step 3

Now that we have our surface looking the way we want, it takes us to

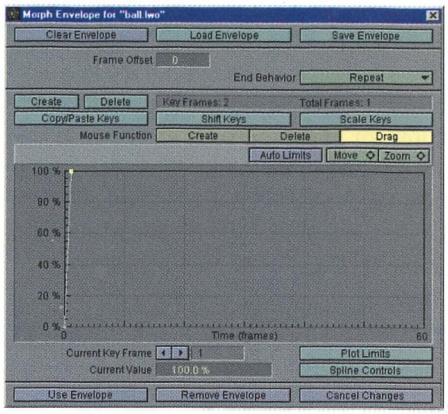


Figure 5 Morph option window.

the next step of softening up those edges to create the illusion of fibers. For this to work, what we have to do is to set up our base shape to morph into our secondary shape. Figure 5 shows the morph option window; you will notice the spline is repeating the morph over one frame. What this achieves is that the geometry now acts out the morph every frame; in conjunction with motion blur it will create the look of our fluffy fuzzball. Figure 6 illustrates the motion blur and anti-aliasing settings used to reach Figure 7.

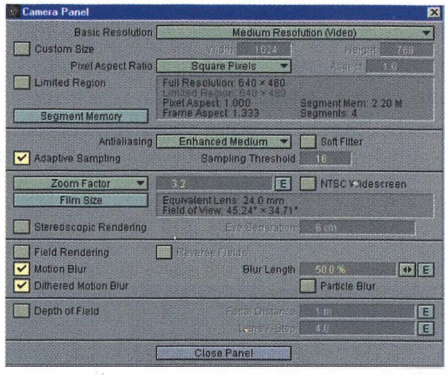


Figure 6 Motion blur settings.

Figure 7 Motion blur applied.

You will note that the motion blur settings are left at their default of 50%. Varying this percentage will give

different results; don't fall into the trap of turning up the blur to try to get longer fibers because, although it does to a certain degree, there are better procedures for obtaining such an effect. If you are looking to make the fibers longer, then take the base shape and create a keyframe at frame 1 which is identical to frame 0. Then, on frame 0 scale your shape by 1.5, again repeating the motion over one frame. The example in Figure 6 uses 1.5 ratio scaling. You will notice our anti-aliasing settings are set to enhanced medium; higher levels will promote a smoother effect. At this stage we will notice the render, although soft in appearance, has a kind of starburst look due to the even emission from the center of the shape. To enhance the look, we will need to break up the geometry to introduce some sort of chaos to the surface; for this we need a displacement map.

Step 4

If we back step for a minute and remember that when we initially created our geometry, we made sure the mesh was triangulated. This is where this action becomes relevant. If we were to apply the displacement to non-triangulated mesh, it would result in rendering errors. Figure 8 shows the displacement map settings we will be using for the purpose of this particular effect, but feel free to try some others and see what you get.

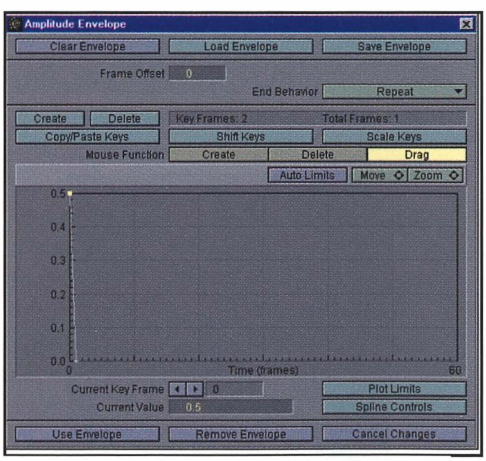


Figure 8 Displacement map settings.

The procedural fractal noise supplied courtesy of NewTek fits the bill perfectly. Once it's applied we should be able to see the points of our shape jittered as shown in Figure 9. What we

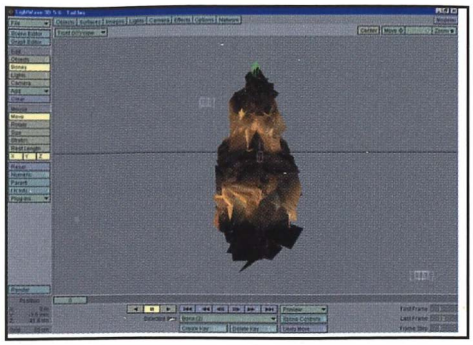


Figure 9 Screenshot of displacement applied.

are looking for here is just enough randomness to mess up the points but still to keep it a recognizable shape. As Figure 10 shows, this really enhances the level of realism to the effect. It can, however, be enhanced further by applying similar enveloping techniques outlined previously (namely, a repeating one frame envelope).

Figure 10 Displacement with fixed amplitude

If we left the displacement at a fixed amplitude, it would simply be the same as jittering the mesh while in modeler when we first created our objects. Figure 10 shows the fibers still emitting as if from the center, due to being cancelled out by the repeating motion envelope we applied earlier. The purpose of the repeating envelope is for better control of root to tip displacement. The repeating envelope invokes changes during the frame resulting in swirling clumps, shown in Figure 11, that look more natural. For the effect to work successfully, we need to use a reverse repeating envelope the same as the motion scaling envelope we applied earlier. You will notice that the envelope not only controls the look better but also decreases the rendering times (compared with a fixed amplitude) due to the length control over the fibers.

Step 5

At this stage the results should look like Figure 11. Although looking convincing, it lacks one final detail to finally enhance the look, namely shadow detail.

Figure 11 Displacement repeating envelope.

To achieve such detail, we should migrate back to our surface attributes for our tail where we will apply a transparency map to the surface. Once again, fractal noise will suit our needs. We are looking for fine grainy detail so we should reduce the texture size to around 5mm. Also, turn the percentage value to 75%, contrast to 8, and small power to 2.5, which will give small sharp detail through our surface. Now that the surface has a level of transparency, it is a good idea to turn on the Double Sided tab for the surface. Also, edge enhancement can be achieved by initiating the sharp terminator option. Figure 12 shows our finished results with the fur effect now appearing in our shadows. The look of the fibers is also enhanced due to the contrast of the transparency that was applied.

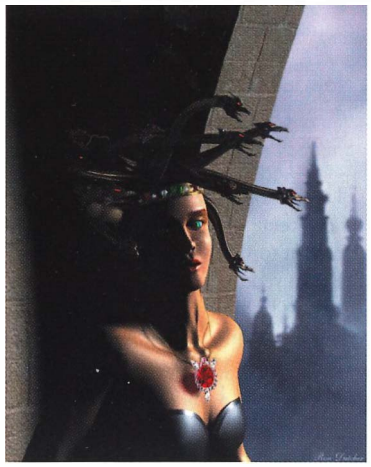
Figure 12 Transparency map applied for tail final effect.

After following these five basic steps, you should have achieved a fur

Continued on page 18

All That Glitters

by Ron Dutcher



If you have spent any time in a jewelry store (If you have a girlfriend or wife you may spend more time there than you like!) inevitably a pushy sales clerk will start lecturing about the four C's of a diamond: Cut, Carat, Color and Clarity. But the clerk will tell you that it is actually the Cut that really brings out the stone's fire. The same is true when rendering photo-realistic gemstones in LightWave. You cannot achieve good results simply with fancy surfacing and lighting; we need to be just as sensitive to the stone's geometry as a fine jeweler.

First, take a look at a typical gem in Figure A. A standard cut gem has three main parts: the crown, the girdle, and the pavilion. These elements need to be carefully proportioned to reflect and focus incoming light back through the crown; otherwise, the light will be scattered out the pavilion, and the resulting gem will be dull and lifeless.

Gemstones 101

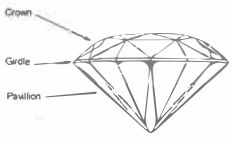
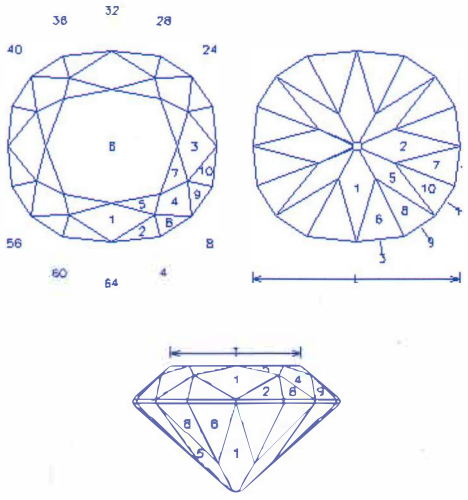


Figure A: Basic gemstone terms

Modeling gems in LightWave is no different than modeling anything else; the more reference material you have, the better. And if you have a few Modeler backdrop images, modeling is relatively easy. The trick here is to find the right reference material.

Most gem cutters work out new facet designs on a shareware program called Gemcad, coded by Robert Strickland. You can download a shareware version and tutorials from http://www.gemcad.net. Gemcad is a DOS based program that displays a gem's top, bottom and side orthographic views. You can design your own original facet designs with Gemcad, but unless you have worked with a faceting machine, this program will be about as intuitive as a Japanese flight simulator. Nevertheless, thousands of public domain Gemcad files are available on the Web, and you can easily load these into the program. Once you have a selected design loaded, do a screengrab (In Windows: ALT + PRINTSCREEN) and paste this into Photoshop. Gemcad displays all the views at equal size, so it is quick work in Photoshop to cut up the screengrab image into the three background files we need. Name these Diamond-top.jpg, Diamond-face.jpg, and Diamond-bottom.jpg accordingly. Or, use whatever file format you wish.

For this tutorial I included three background images of the famous Hope Diamond. You can download the images from my website. http://www.mbs-clinic.com/hope.html



Top, bottom and side background images of the Hope diamond.

Now, open LightWave's Modeler, and click the Display tab. Click on the Backdrop button to open its Option Panel. In the Y Image slot load the Hope-top image, leave the Center requesters alone, and for Height enter .05. In the Z image slot, load the Hope-face image with the same settings as above and hit OK. Zoom the grid in or out so that the backdrop images fill the views.

Now we are ready to model. Type the plus sign(or click on the Polygon, Points button) to activate the Points tool. In Modeler's Top view, position the crosshairs on the facet angle directly below the diamond's center. In Modeler's Face view, position the crosshairs on the facet edge directly over the 1 in the Backdrop image, and hit Enter. Back in the top view, moving clockwise to keep the vertices facing outwards, position the crosshairs over each facet angle and hit Enter for each point. Now hit the "p" key or click on the Make button to make our first polygon. Now go back to the Points tool, and position the crosshairs on each facet angle for the crown. The facets will not match up perfectly in the Face view, but that is all right. We just need the Face view to find the correct height. Once you have all the points created, deselect them. Now, in the Top View, select points for each facet and hit "p." Watch the OpenGL preview window each time you make a polygon to be sure it is aligned correctly. If a facet is facing the wrong way, select that polygon and hit "f" to flip it back with the others.

Once you have all the crown polygons made, select all the points that make the top of the girdle in the Face view, and hit "c" to copy them. In a new Modeler layer hit "v" to paste these points. Hit the "t" key or click on the Move tool, and while holding down the "CTRL" key, move these points down to the bottom of the girdle in Modeler's Face view. Hit "x" to cut these points, and back in the original layer, hit "v" to paste the points. Now we have the correct points to create the girdle. Select the points, for each girdle facet and hit "p" to make them just like we did for the crown. This is a little more tedious, because you must deselect the points on the opposite side of the gem. You should have something like Figure B on the next page.

Since we are finished with the gem's crown and girdle, go back to the Backdrop Option Panel; and in the Y Image slot, load the Hope-bottom image. This will be our reference for modeling the pavilion element of our diamond. Just as with the crown, create the points at the facet angles in the top view, select the correct height in the Face view, and hit Enter to make each point, then select the points for each facet and hit "p". To make things a little easier, you can select all the polygons in the crown and hit "-"(the minus sign) to hide them from view and keep them out of the way. To bring them back, just hit the Backslash key. If we made all the facet polygons correctly, we should see the Hope diamond in Modeler's OpenGL window.

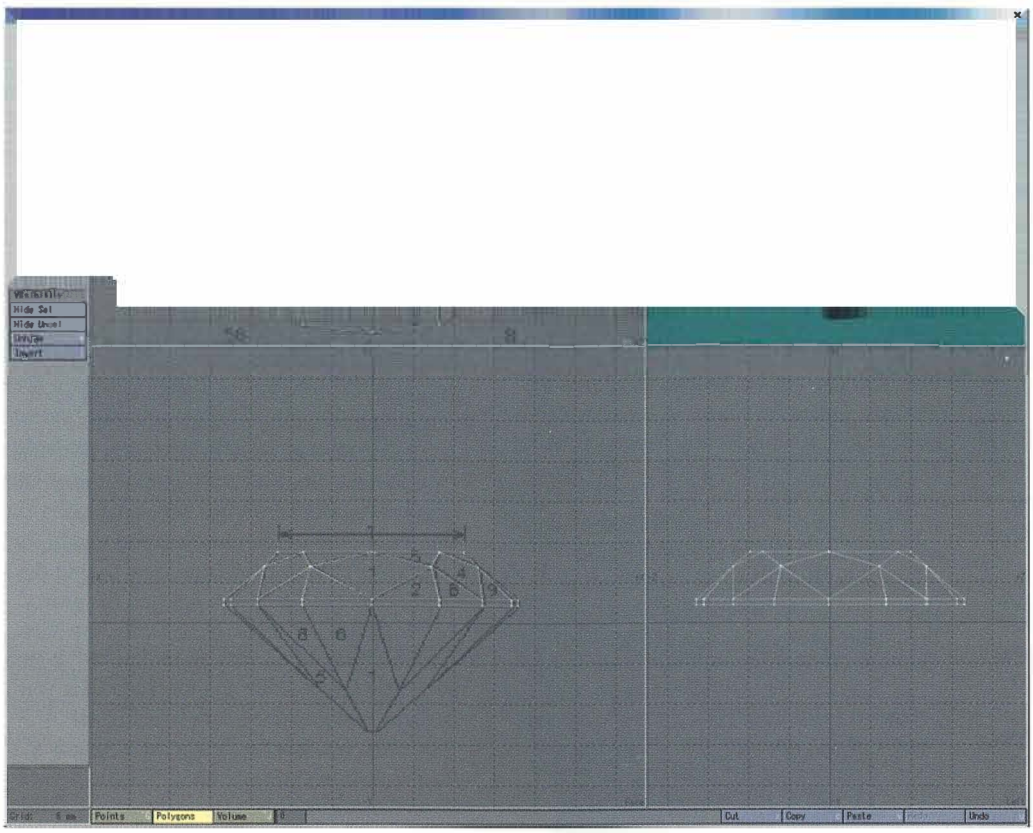


Figure B: Modeler showing copy of Hope diamond.

To be safe, hit the "w" key while in the Polygon mode to bring up the Statistics. Make sure you have no polygons selected and click on the "+" beside the Non-planer slot. If no polygons become selected, you are a much better modeler than I am. These facets need to be precise and if we made even a slight error in plotting the points, the polygons will become non-planer. For still images this might not make much difference, but in an animation the polygons would flicker and look stupid. To fix this, hit "T" to triple the selected offending polygons. This will fix all our problems.

With no polygons selected, hit "q" to bring up the Surface panel. Name this surface "Hope-Out". Pick a white color (R243,G243,B243), Diffuse 10%, Specular 100%, Double-sided and Smooth both Unchecked, and click on the Apply button. Actually the Hope diamond is a rare, deep blue diamond; but for now, imagine that it is a typical clear diamond.

In LightWave, to correctly render refraction, you need an air layer in your transparent objects to tell the rays when they are leaving the transparent material. To do this for our diamond, make sure Modeler's Polygons Mode button is selected and hit "c" to copy all of our diamond's polygons, and in a new layer, hit "v" to paste them. Hit "f" to flip all the polygons. Hit "q" again to bring up the Surface Panel, and with the same settings as before, rename these polygons "Hope-Air." Hit "x" to cut these polygons, and back in the original layer, hit "v" to paste them. Hit the "m" key to bring up the Merge Points Panel; and with the Automatic option selected, Hit the OK button. Now we have a diamond that will refract correctly; so save this object as HopeDiamond.lwo, close Modeler, open up Layout, and load the Hope diamond object we just modeled.

Now we are going to add some polish to our diamond. Open Layout's Surface panel, and in the Basic Parameters, enter these settings for the "Hope-out" surface:
Surface Color: 243, 243, 243
Luminosity: 0%
Diffusion: 10%
Specular Level 100%
Glossiness 55%
Reflectivity 0%
Transparency 95.5%
Color Filter checked
Refractive Index 2.417

Hit Render to see the surface in the preview. Select the Hope-Air surface, click on the rendered preview we just made to copy these surface settings, and set the reflectivity to 85%. Also, change the refractive Index to 1.0003, Air's Refractive Index. 2.417 is a diamond's actual Refractive Index.

Now let's make a DeBeer's advertisement. Position the camera above the diamond, facing down upon the table facet. Don't position the camera too close, but let the diamond fill most of the screen. The closer you move the camera to the gem, the longer it will take to render. Also, rotate the default Distant light so that its pitch angle is 90 degrees, while Heading and Bank are both set at zero. In the Lights Panel, set the Ambient Light to 0%. In the Render Panel, activate Reflection and Refraction. You probably know that when you render reflections and refraction you are going to take a big hit in your render times. To speed things up a bit, set the Ray Recursion level to 5 instead of the default 16. This will make a big difference in your render times, and since we are working with faceted gems, we shouldn't see much (if any) loss in quality. Typically, I see the greatest loss in quality with Ray Recursion Limit reduction when working with smooth, curved, transparent and reflective objects. Now hit F9 and your image should look something like Figure C. Not too bad, with no environment to reflect or fancy lighting.

Figure C: Basic rendering of Hope diamond.

With this scene, we have the light coming straight down to enter the crown, but the specular highlight on the table is a little too strong. To solve this problem, select the Hope-Out surface in the Surface Panel, and click on the Advanced Options. In the first Shader Plug-in slot select the LW_FastFresnel shader, and click on its Options button. Uncheck everything except for Specular; set that to 100%. Click OK, go back to the Basic Parameters, crank the Specular down to 10%, go into the Objects Panel, and click on the Save All Objects button to save our texture settings. Now hit F9, and our diamond should look a little better.

If you visited the Smithsonian to view the real Hope Diamond, you would likely be

surprised to see how blue it is. Geologists believe this diamond was created near the Earth's core nearly a billion years ago, and later brought to the Earth's surface in a volcanic eruption. If you want to recreate an authentic-looking Hope Diamond, try changing the Hope-Air surface color setting to R125,G125,B218.

On my little notebook computer, it took 3 minutes, 11 seconds to render this scene. That's not too bad, but if we start adding other gems, or gems with more complicated facet designs to the scene, the times will become astronomical. So, let's do something to help this problem. Included with this tutorial, or found on my website, is a file called "RD-reflect.jpg." LightWave comes with some good reflection maps, but these work best for metal and won't help us much with gemstones.

RD-reflect.jpg reflection map.

Go to Layout's Image Panel and load the RD-reflect file. In Layout's Surface Panel, select the Hope-Air surface, and click the Reflection Options. Set the Reflection Type to Spherical Map, in the Reflection Image slot, select our RD-reflect image, and click Close Panel. Hit F9 again and BOOM; the scene now renders in 23 seconds. Also, the resulting test render is not much different in quality from our first render. All you need is the right reflection map, and you can save yourself a lot of headaches.

Now that we have a Diamond, try some other precious stones. Rubies are always popular, so go back to our Hope-Air settings and change the color to R138,G0,B24 to bring out that rich pigeon blood color. Next, select the Hope-Out surface and change its color to the same, and change its Refractive Index to that of a ruby, 1.76. The highest quality rubies come from Burma. One difference between these and lower quality rubies are the specular highlights. The lower grade rubies reflect a whiter glassy highlight, while the higher grade ones reflect a deeper red highlight. To simulate this, check or uncheck the Color Highlights button, depending on which look you are shooting for.

Sapphires are made of the same mineral as rubies and share all the same traits, except for color, of course. So to create a sapphire, simply change the color to R32,G32,B113. You will probably want to change the background color to something lighter to get more contrast.

You can create several interesting effects by applying filters or image maps to the Hope-Air surface. Try experimenting with LightWave's Thin Film and Interference shaders. Brenden Mecleary's RealiTools II (http://www.dynamic-realities.com) includes 2 shaders, Iridescence and Translucency, that work well with gems. See Figure D.

Figure D: Effects of various shaders and filters applied to Hope-Air surface.

One problem that I often come across is lighting a scene that has diamonds and other darker stones, like amethyst or garnets, together. If you light the scene to make the diamonds look good, the darker stones look dull. If you light up the darker stones, then the diamonds wash out. You can solve this problem in two ways. The first way is the plug-in approach. Apply Worley's Gaffer (http://www.worley.com) to the darker stones to boost the lights hitting them. Also, to boost these darker gems, try experimenting with placing colored lights inside the gem. Position these lights just passing through the table facet's plane and shining onto the pavilion.

The sans-plug-in method is to take a negative Intensity spotlight, positioned just in front of the washed out diamond. Start with a Light Intensity of -50. Set the Intensity Falloff to reach a little past the pavilion, and adjust the negative value to suck up as much extra light off of the diamonds as you like.

Take a look at any of the online jewelry stores on the web, and you will notice that all of their photos show gems with a just a faint diffused shadow--just enough of a shadow to give the photo some depth without becoming distracting. In LightWave, even the shadow mapped lights create hard annoying shadows. An old, but reliable, trick to get nice diffused shadows for your gems is to set up your key light just right then divide its intensity in half. Then clone the light and turn off the clone's shadows. You can now control the strength of the shadow by increasing or decreasing the clone's intensity. For the clone light, I like to use an Area light. Since it isn't casting a shadow, it renders rather fast, and you still get that nice Area light feel. For the Shadow light, I like to use a spotlight with shadow mapping. Crank up the Shadow Map Size to as big as you can afford. When working with gems, I usually work with a size of 3,000. If I make it any bigger, my system will start to bog down as it pages for virtual memory. I also like to make the shadow really fuzzy to look like the DeBeer's photos; I set the Shadow fuzziness to 10 or more. If you are working with small gems, or necklace chains, and you make the shadow too fuzzy, you will wash out your shadow completely; so start small, and work your way up.

Next time you are watching TV late at night, and surfing through the channels, stop on one of those shopping channels for a minute. Eventually they will show some huge, gaudy jewel with the unbelievably low price of $59.99!!! What really catches the eye for the people who buy these things is the fantastic sparkles. With good lighting, camera positions, and star filters on the camera lens,

Continued on page 18

Sahara

by Tareq
Character Modeler/Animator
Foundation Imaging

The name of the character on the cover is Sahara. She is a belly dancer that I am working on for a personal project called *Sahara and the Beast*. At this time, I am just building the models and it will be awhile before the animated short is done. When I was asked to do the cover for *Keyframe* I decided to make it a challenge and to try to combine as many programs and plug-ins as possible to pull off the image.

I started out by modeling Sahara in LightWave 5.7. The model was made using MetaNURBS. The female was initially built in about 14 hours. Tweaking the model for Sahara took a lot longer. The head was built separately and added onto the body. Once the body was done I started adding on clothing. I wanted the character to really hold up at high resolution so I NURBed everything on her. The skirt was modeled flat out so I could texture it planar down the Y axis. That way, the soft body dynamics can deal with the skirt's deformation. I quickly ran into the point limit for a metaformed object at only one level, so I finished my model off in LightWave 6.0.

After the model was built, I wanted to build the bone structure and do the soft body dynamics in Messiah. I reverted the cage back to 5.7 so I could bring it into Messiah. I built the bone structure in Messiah pretty fast. It's so easy to set up bones with the interactive bone feature.

After the setup was done, I moved on to the soft body dynamics for the dress. After a day of playing with it, I gave up and opened up Motion Designer for the dress dynamics. I positioned the character within Messiah and saved off a linked scene. But, you might be saying..."Tareq, I thought you wanted to use L6 to render it in. Messiah doesn't work with it!" I know...so I saved a transform off in Messiah of the posed character. I brought the character back into L6 modeler and made the posed model a delta morph of the character. That way I was able to use the morphblender in L6 to deform the object. After that was done, I did some dynamic tests in Motion Designer. I had

The image was rendered on a P2 400 with 512 MB of RAM.
Polygon Count: +200,000

Software:
LightWave 5.7
LightWave 6.0
Messiah 1.54
Motion Designer 1.1
Shave and a Haircut 1.27
Photoshop 5.0

Messiah screen

some success with that and saved off another transform with the dress for some final tweaks in modeler. You must be thinking..."Damn Tareq, that's a lot to go through." Well, it's not over yet.

Next, I made a dummy object of the head with collision spheres in it so I could make hair for the character in Shave and a Haircut. After trying many things with L6 and Shave, I decided to do the character and the hair as separate passes. The shot was done with two area lights in L6. I was only able to bump the Subdivision level up to 2 before it gave me the memory allocation error. So much for better results in L6. The hair was rendered separately, because Shave and a Haircut cannot be rendered in segments and does not work well with area lights. I had to render the hair at a slightly lower resolution than the body because of the single segment restriction with Shave.

After rendering my two passes, I went into Photoshop, composited them together, and made final tweaks to the image. It was a long and hard process. What made it more difficult was trying to make it work at print resolution. In the end I was really pleased with the results. The combination of LightWave 6, Messiah, Motion Designer, and Shave and a Haircut really paid off. I recommend you buy them all!

Shave and a Haircut screen

Continued from page 15

even the worst quality gemstone will sparkle. To recreate this simple-rotating-sparkling-gem animation, first set up the scene's camera, lights and the gem just the way you want. A darker background works best. Add a Null object and parent the gem to the Null. Now set the Null's rotation to the speed you want. If you want the animation to loop, Move the Slider Bar to Keyframe 60 (or for however long you wish), rotate the Null's Heading 360 degrees, and hit Enter twice. If you don't already have OpenGL turned on, select it in the Layout View Options. Slowly move the Frame Slider forward. Notice which frames have facets showing strong specular highlights and jot these down in your notes. Add a spotlight and rename it Flare. If your gem is a diamond, you will want the default pure white color light. If you are working with other stones, experiment with different colored lights, but generally near white works best.

Change the Light Intensity to 0%. Check the Lens Flare box and click on the Lens Flare Options button. Uncheck Red Outer Glow and Central Ring, and in the Star Filter, select the 4+4 Point star. In the Flare Intensity Envelope, enter the keyframes that coincide with the specular highlights that you jotted down earlier. After you have a good envelope, close the panel and go back to layout. Parent the Flare light to the Null, and in keyframe 0, position the light just in front of your target facet that will reflect the highlight. Move the slider bar to where a sparkle should show up and hit F9 to see a test render. You will probably need to tweak the envelope several times before you get

Figure 14 Keyframes in our animation showing achievable dynamics.

Continued from page 12

effect that looks real enough to keep you warm over the winter. Now that we are familiar with the effect, let's create something more familiar that will also let us investigate how much more we can push the effect to meet our needs, namely skeletal deformation and dynamics.

Step 6

With geometry in place and textures applied, we should have something that resembles Figure 12. From here we want to start setting up our skeleton within the mesh. Figure 13 illustrates what we are looking for, namely, a single chain that runs the length of our tail object.

What to keep in mind when setting up your skeleton is that the skeleton links by default originate at frame zero Thus, its rest position is frame zero; but what we need to do is to set the skeleton in place to control the geometry's shape on frame one. This is because our morph envelope changes the volume of the mesh from frame zero to one. What you have to do is to orient the links on frame one, but create a key once positioned on frame zero, which will result in setting the rest position on frame zero. On completion of these procedures, we should now have one fur tail ready for animating. What our animation will show us is the level of dynamics that is achievable from what is in essence a motion blur effect. Creating a couple of keys that swing our tail chain from side to side will set us up for rendering to see the effect. Figure 14 shows three frames from our animation. As the tail swings, the fur effect appears to inherit secondary motion from the motion of the tail. It looks like the tail's fur is shorter in the leading direction of the swing and longer on the trailing side. This is due to the nature in which the motion blur works. Specifically, since backtracking previous frames (the higher the anti-aliasing pass the smoother the effect on extreme motions due to more pass information for the motion blur) are proportional to the movement of the object, it causes dynamic reactions to the motion of your object. This could be reversed if our morf order was reversed.

There are many other neat effects that can be applied to further enhance your furry creatures, like applying the

something you like, but this simple lens flare can create sparkles that would make even home shopping channels envious.

Ron Dutcher is a radiologist, working in a Japanese orthopedic clinic. In between taking X-rays, he creates medical animations and does some freelance graphics work. Contact him at rd@mbs-clinic.com

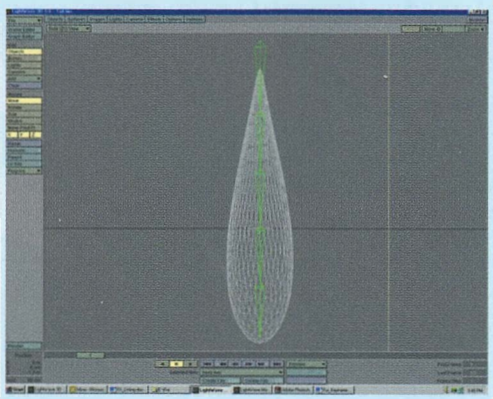


Figure 13 Tail skeleton applied.

Continued on page 21

LightWave Imaging for Print

by Tiffani Banaszak

In this tutorial, I will explain the best way to prepare your LightWave renders for print. As the Art Director for Keyframe Magazine, this is a task I do with every issue, and it is a subject I am asked about quite often.

Preparing Your Image

This process is easiest if you are aware at the time of rendering that your image will be needed for printing. I can't tell you how many times I've asked a writer to submit a "print res" image for a cover graphic, and they've had to re-render it (much to their dismay). Rendering an image large enough for our cover is no small task!

The questions to ask the graphic artist when preparing to render your image are:

1. What is finished print size, including bleed?
2. What is the printing dpi?
3. What file format do you need it in?
4. Are there going to be other graphics, text, etc. used with or on top of the image? If so, where? If you can, get a sketch or accurate measurements.

These are questions that will be familiar to a graphic artist, but their *answers* may not be helpful to a LightWave artist. We print-heads speak in dpi and print res, and you animator types speak in film specs and pixel resolution.

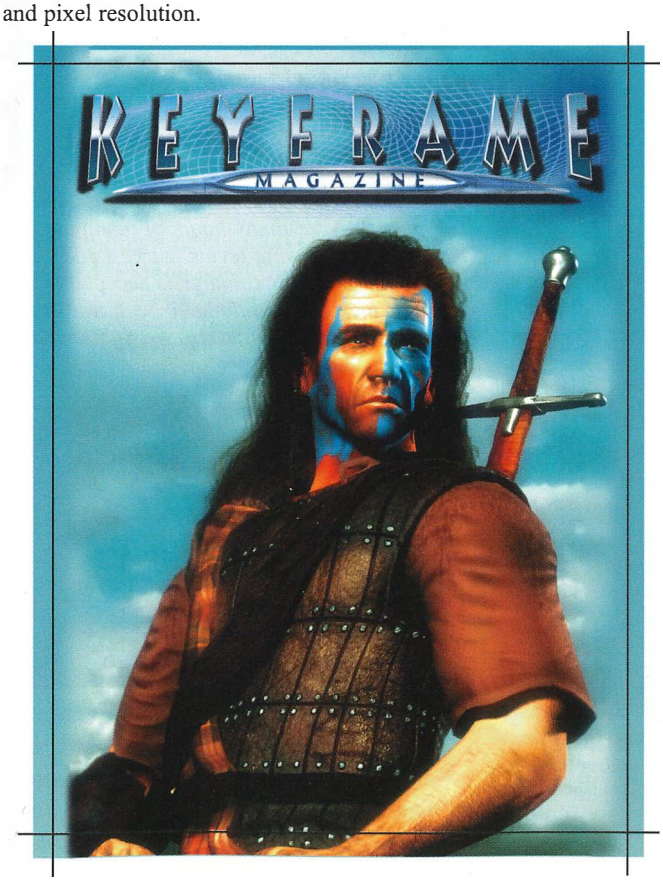


Figure 1: The cover for Issue 16 showing the bleed and crop lines.

Let's use Keyframe's cover for an example. If you take out your ruler and measure it, you will find the cover is 8 inches across and 10.75 inches high. The image we require for the cover must be 8.5 inches across and 11.25 inches high, to account for the *bleed* or trimming of the edges. The standard bleed size is .25 inches, as shown in Figure 1, so keep all detail a *minimum* of a half-inch away from the edges. You never know where that trimmer will slice!

The easiest way to convert graphic-speak to animator-speak is this: take the finished print size and multiply it by the printing dpi for length and width. That is your pixel resolution. Taking our cover size, you would multiply the width (8.5) by the dpi (200) to get 1700. Doing the same for the height, you get 2250. So the image you need to render for the cover must be 1700 by 2250 pixels. Simply set your render to this custom size, and voilà! You have a cover image.

The image format is another important issue. Generally, I receive images for the magazine in .tga format, which is fine. But not all versions of Photoshop can open .tga files. Make sure your graphic artist can open the file type you are planning to supply! I do not recommend .jpg or .gif images at all, since they are usually low resolution and will not print well, as shown in Figure 2.

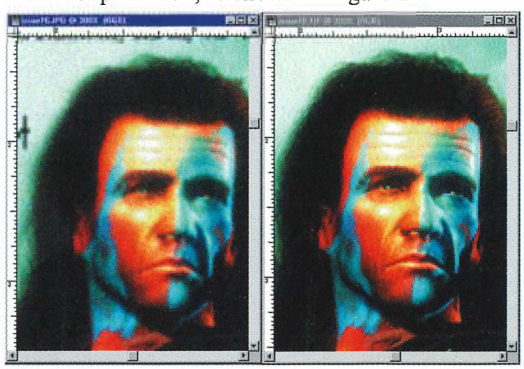


Figure 2a and 2B: Comparison of a .jpg and a .tif image. The .jpg is on the left, and you can see a lot of image loss compared to the .tif on the right. This image was blown up 300% for visualization purposes.

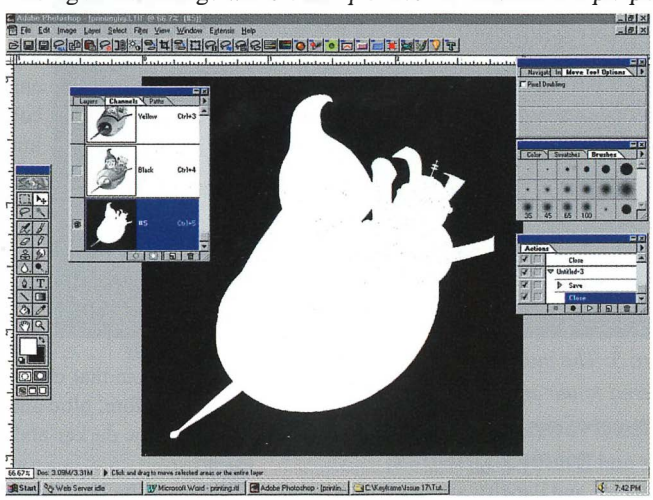


Figure 3: The alpha channel is used for creating tight outlines, called a mask, around part of an image. This is handy when objects have small parts, hair, or particles that are difficult to mask out by hand.

If you have transparencies or alpha channels to save, .tga works best, as shown in Figure 3. But all images will be converted to .tif for printing. Let the graphic artist do all file, size, and color conversions! It saves headaches all around.

Taking into account the placement of other images, logos, or text on top of your image is important. Figure 4 shows you the original image submitted to us for Issue 14's cover. As you can see, it is not the proper scale or width/height ratio. Flash Film Works re-rendered the image so it would fit our requirements, as shown in Figure 5.

Figure 4: The original image from Deep Blue Sea.

Figure 5: The image we used for the cover was rendered to our specifications, allowing proper placement of our logo, barcode, and text.

Color and Density

Color and density are other important factors to consider when rendering your image. Images originally created for film and video tend to be darker, with deep shadows and fewer areas of bright color. This is great for dramatic purposes in a transmissive medium (one which provides its own light) but does not look nearly as good in the reflective medium of ink on paper. It is important to keep in mind that the phosphors in your monitor and TV produce brighter, more vivid images. Paper and ink, on the other hand, will absorb light and create a darker-than-normal image. A rule of thumb I use when trying to judge how an image will look when printed is that if it looks just a *bit* washed out on my monitor, it will print at a good density.

The darker the image, the less detail will show, especially if there are areas of high contrast (Figure 6). Lightening the image in Photoshop produces a washed out image. More detail is apparent in the dark areas, but it becomes very faint in the lighter sections (Figure 7). A good graphic designer can get an acceptable image, Figure 8, but it is not as good as it could be.

Another consideration when printing images is the conversion from RGB to CMYK color space. All LightWave images will be saved by default as RGB, which is the method monitors, television sets and other viewing devices use to display color. But all printers require images in CMYK, which are the four ink colors used to create printed images (Cyan, Magenta, Yellow, and blacK).

In general, a CMYK image will be less vibrant, and there can be extreme color differences, most evident in the blue spectrum. In Figure 9, you can see the Red, Blue and Green channels, showing where the colors are located in the image. Figure 10 shows you the same image in CMYK. Looking at the blue channel in Figure 9, you can see that much of the image shows up here. Comparing this to the Cyan channel in Figure 10, you can see that there is less image in that channel, and more in the Black. Therefore, blue and many gray colors tend to become darker and less vibrant (Figure 11).

The best blue to use when surfacing or coloring the objects in your scene would be one with a green cast, like a turquoise or teal color. This is closer to the Cyan used in printing and will show up much better. Most reds and greens will reproduce more accurately than the blues.

Figure 6: The original render with no adjustments. You can see there is very little detail.

Figure 7: The same image, with a simple brightness adjustment in Photoshop.

Figure 8: After many hours of manipulation in Photoshop using levels, curves, and brightness, the final image is acceptable, but not as nice as it would have been if it had originally been rendered at a brighter setting in LightWave.

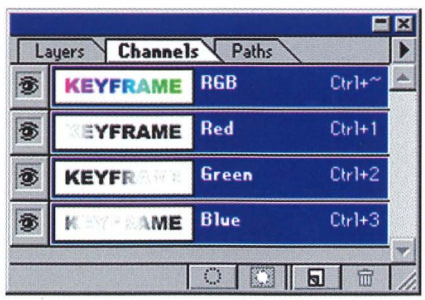


Figure 9: An RGB image showing each channel.

And Finally

Ask to see it before it prints! As a rule, I

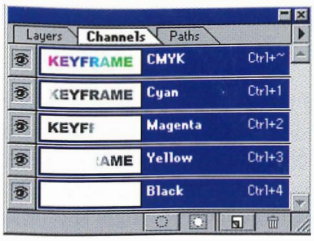


Figure 10: The channels of the same image as in Figure 9, but in CMYK color space.

email a copy of our cover to the artists, both for their enjoyment and their critiques. The calibration of your monitor will be different from the artist's, but at least you can tell if a vital part of your image was cropped off or covered with a logo. As the artist, you have the right and the obligation to make sure your work is used as you want it.

If you require copyright, trademark, or other such credits with the image, be sure to inform the user. Whether the credits are your own or someone else's, they need to be included. Many of our covers and other images are from well-known television shows, movies, and commercials. The true owner of these images is not the artist! For example, these *Roughnecks: Starship Troopers* images were created by Foundation

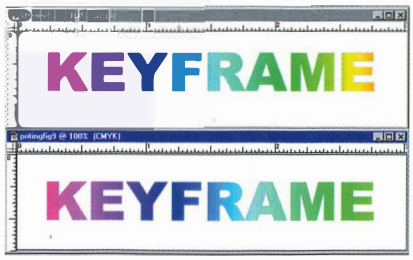


Figure 11: A screen grab of an RGB image next to a CMYK image. Notice the color disparities.

Imaging, but are owned by Sony (Figure 12). Foundation Imaging had to contact Sony to obtain permission, as well as the required verbiage that must accompany each and every image.

As a magazine dedicated to LightWave and the works of its artists, we love to receive images for our gallery and cover. (C'mon guys, send us stuff! We want to start up our gallery again!) Following these guidelines will ensure a smooth transition from animation to print, and knowing what is required in advance will put your images at the head of the line.

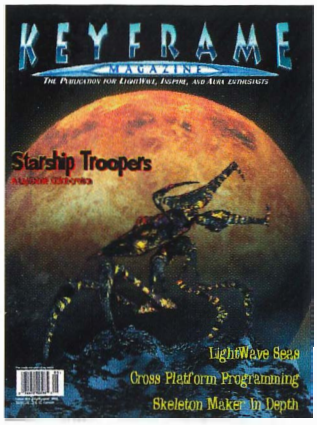


Figure 12: RST image created by Foundation Imaging, copyright Sony.

Tiffani Banaszak is the Art Director for Keyframe Magazine. She can be reached at Keyframe1@aol.com. Please visit our web site for specifications for submitting cover and gallery images, tutorials, and product reviews.

Continued from Page 18

morf surfaces tab which will allow varying your morf targets to have unique surfaces. This can envoke a multitude of different effects, one of which is mainly used to vary root to tip coloration. Another neat trick to try once you have mastered the technique is to tweak the points of your morf target while in modeler to further aid in styling a desired look. Backlighting attributes can be achieved through application of Gaffer to give the surface a element of translucency. All in all, a very effective effect can be obtained from the basic tools in your toolbox; so let your hair down and don't be afraid to try different variations with the settings. Before long you're sure to become a master digital stylist. See if you can spot where the technique has been used in Figure 15.

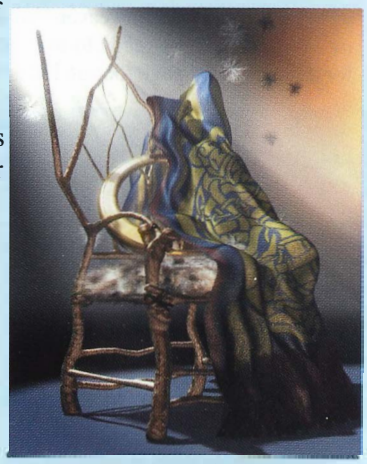
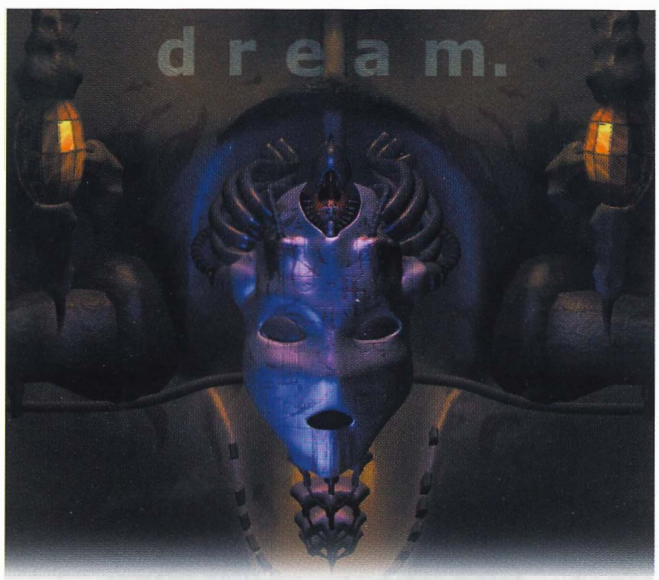


Figure 15 Techniques put into practice.

Graham McKenna is a founding artist of Axis Animation. You can contact him through his email, graham@grahamprismatic.free-online.co.uk, or website, www.grahamprismatic.free-online.co.uk



3D Exchange
Animation
Training

Premier Authorized
Training Center

NewTek

www.3dexchange.com • (510) 747-1540

Character Animation for the Web

by Marc Williams

Web-based character animation has become a hot topic in the animation industry. This case study will take you through a complete web project here at Spydaman Studios with LightWave, and will, I hope, broaden your skills so you can take on web animation projects for *any* client.

The first step in any commercial project is a brief from the client. I was told to create a small *.gif animation file which can be used as a link button. I was also told to use the company mascot, a lamp, to be animated in a looped cycle. The client's company name, Lamps & Stuff Ltd., needs to be next to the moving picture.

I planned to create the looped animation in LightWave. I decided that the lamp should be hopping over about 40~50 frames. As the animation needs to loop seamlessly, I need the start and end frames to be very similar in terms of position and lighting.

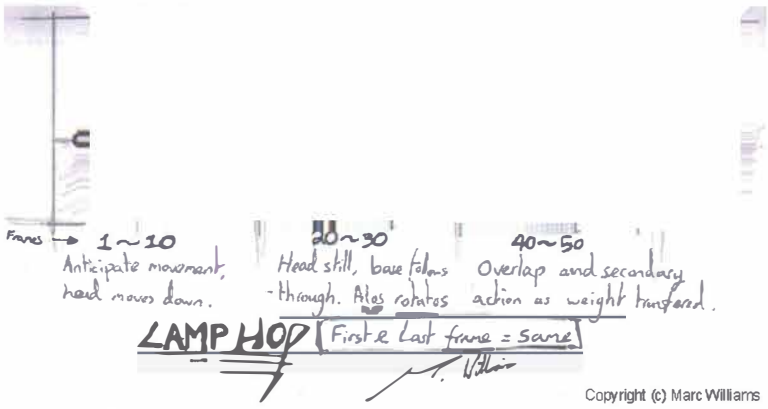


Figure 1: Storyboard sketches of lamp animation.

I modelled the lamp in both low-res and high-res versions. This saves a lot of computing time with slower processors and can give better feedback. After model checks and test renders, I set up the IK for the lamp. I positioned the low-res lamp model, added and tweaked IK, and tested all conceivable positions and poses over two hours.

Once I had the correct IK setup, I began animating the lamp. First I got the jump timing right, then I added all the touches that make it more real (follow-through, secondary action, anticipation, etc.). I then had the tricky task of making the loop. This is difficult because the lamp moves across the screen. Because the action is looped, I need the start and end frames to be roughly the same, so I had to move the camera along the x-axis to keep the lamp in the center of the stage. The viewer's eye will not detect this camera move and the lamp will look like it's hopping on the spot.

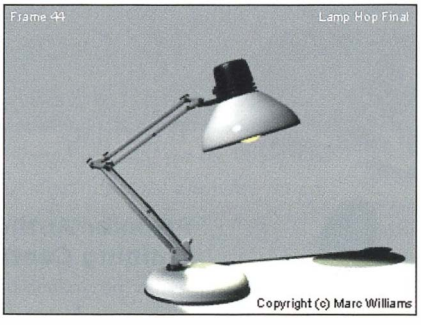


Figure 2: Test render of lamp.

I rendered a test animation of this in 320x240. When I was happy with the movement, I set about adding textures and lighting the stage. First, I replaced the lamp low-res objects with the high-res ones. I then added the bulb and a floor or stage. I made the background color and the stage the same gray colour. Then I added a spotlight for the lamp, parented in the usual way. I then tweaked the values of the light until they were satisfactory. The textures of the lamp were basically shiny white metals and black rubber from my texture library.

For the final render, the camera resolution was set to custom of 40x40. This produces a small animation which is just right for the client's needs. After the test animation looked good, the same files were rendered as separate *.jpg frames.

Figure 3: Single frame of final image sequence, magnified 4X.

I imported the frames into Adobe ImageReady (This is now a component of Photoshop 5.5 and is no longer sold separately.) and then optimized them at a low gif setting. I then deleted each second frame. This cuts the files and frame size in half without a big loss in quality. I then saved the image sequence as a single gif file which can be added to the web. Other programs can also perform these functions, including Animagic (rtlsoft.com/animagic) or CoffeCup (www.coffeecup.com/animator).

I still needed the company name next to the animation, so I used Photoshop to create the side picture.

Figure 4: Side picture to enclose animated GIF button.

I then used Dreamweaver 3 to add the side picture and the gif file into one HTML file ready to use. Following is part of the HTML code for a simple web page that includes the gif and side picture.

```
<div id="Layer1" style="position:absolute; width:200px;
height:8px; z-index:1; left: 33px; top: 30px">
<img src="Button.gif" width="200" height="53"></div>

<div id="Layer2" style="position:absolute; width:30px;
height:36px; z-index:2; left: 39px; top: 36px">
<img src="Lamp.gif" width="40" height="40"></div>

<div id="Layer3" style="position:absolute; width:200px;
height:71px; z-index:3; left: 256px; top: 23px"></div>
```

This method of web animation can be used for any animated content such as logos or animated characters that 'host' the page. The possibilities are endless. So have fun, and happy animating!

Marc Williams is Head of Animation and Design at Spydaman Studios. You can contact him through the studio's website, www.williams12045.freeserve.co.uk

Skeleton Modeling

by Sean Moyer

I have a skeleton in my closet, and his name is Ray. And like most such skeletons, he hides my dark secret: sometimes I cheat. On occasion, I have been known to model a sports car, and to leave the threads off bolts that hold the oil pan in place. Sometimes I even skip the bolts. And if the client doesn't want to show the bottom of the car, I'll even leave out the oil pan. Sacrilege! But the way some people model, you'd figure they were getting paid by the polygon. Reality check: your time is money. Get the most out of it by using some of the quick and dirty tricks of the modeler's trade.

Ray started life as a pair of images I found on the net: a front and a side view of a skeleton. Loaded into Modeler as background images, these provided guides for the basic proportions of a skeleton. I used supplemental images to figure out some of the trickier bone structure, among them a set of medical posters and a website that displays human anatomy from X-rays. If you are interested in modeling your own skeleton, I recommend picking up some good source material. I searched the web for skeletal anatomy, and I picked up a book called "The World's Best Medical Charts" (ISBN 0-9603730-5-5). Ray's ultimate purpose is entertainment, not medical, and so he owes more to Harryhausen than to the Visible Human Project. He's a work in progress, seven hours and counting.

I modeled Ray from the top down, literally, starting with the skull stand-in (a detailed skull is a project unto itself), and beveling and smoothshifting down from there. I accomplished much of Ray's construction with very simple tools (move, rotate, drag, stretch, knife, and smoothshift), albeit used in ways that are not always implied by their names. I designed the model to be metaNURBed.

For the purposes of this tutorial, I'm going to focus on two parts of the skeleton that presented different challenges and demanded different modeling approaches. In order of difficulty, they are: the Humerus (upper arm), and the Sacrum/Coccyx (tail bone).

The Humerus is your standard-issue bone, skinny in the middle and knobby on both ends, but even a simple bone like this requires a bit of special attention. To begin, I placed a box that approximated the length and volume of the bone (not including the knobby ends), and rotated it into roughly the correct position. Two quick swipes of the knife, a bit of dragging, and the basic shape is laid out, Figure 1.

The challenge posed by this bone is how to elegantly branch the

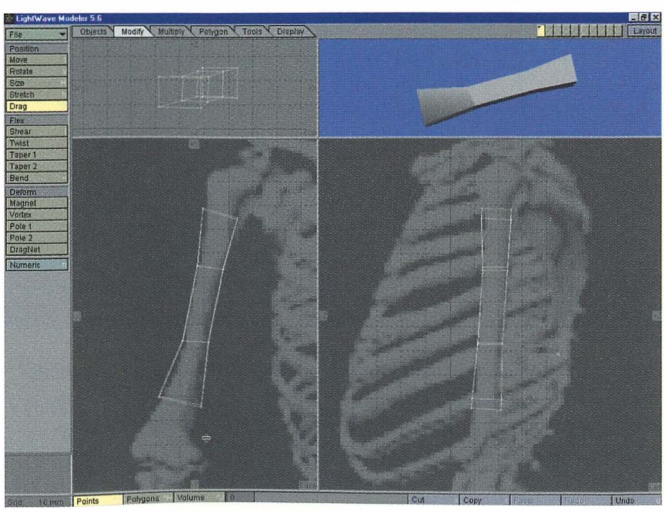


Figure 1: The initial geometry of the Humerus.

two bony knobs at the bottom of the Humerus from a single quad polygon at the end of the bone. First, we need to pull a little more geometry off that quad. The easy answer is a bevel, but the bevel tool has four input fields, and the last thing you want to do in a quick and dirty modeling session is think. So use smooth shift instead, with the offset set to zero meters. Then move the new offset polygon into place. Two hotkeys, smooth shift and move, and your new geometry is where you want it, without going through a try/undo try/undo cycle with the bevel tool.

Keep the smooth-shifted polygon selected. Switch over to the stretch tool (keyboard shortcut h). In the X-Y window, slap the stretch tool in about the middle of the selected polygon. A quick swipe of the mouse left and down, and we've collapsed the quad by stretching both dimensions to zero. This forms the branch from which the two bony knobs will be created, Figure 2.

Figure 2: The collapsed quad.

Some quick cleanup is needed before we move on. Kill the selected polygon (keyboard shortcut k) and merge points. I used an absolute value of 1mm, your mileage may vary depending on the scale you've chosen to model the bone. This gets rid of unnecessary points, and prevents the creation of a 2-point polygon.

Select the two polygons that now face in the direction of the knobs. Now two quick bevels, one with a negative inset, and then the same bevel with a positive inset, and we have the rough shape of the knobs, Figure 3.

Using the background image for reference, drag points to get the rough bevels to more closely match the shape of the bone. Remember that with metaNURBS, the cage geometry is more exaggerated than the final product. Don't just line the points up on the edge of the bone image and expect it to look right. Also, don't be afraid of a little overlap. Where the two knobs touch each other, you may need to drag the rough geometry of the knobs into each other a bit for it to look right when metaNURBed. Think organic. Symmetry is your enemy. Remember to tab back and forth between the rough and the metaNURBed version for an indication of the true shape, Figure 4.

Next stop, a couple of funny little bones in the nether regions called the Sacrum and Coccyx. Together, they comprise the more familiar tail bone, and I've chosen to model them as one piece. Take a

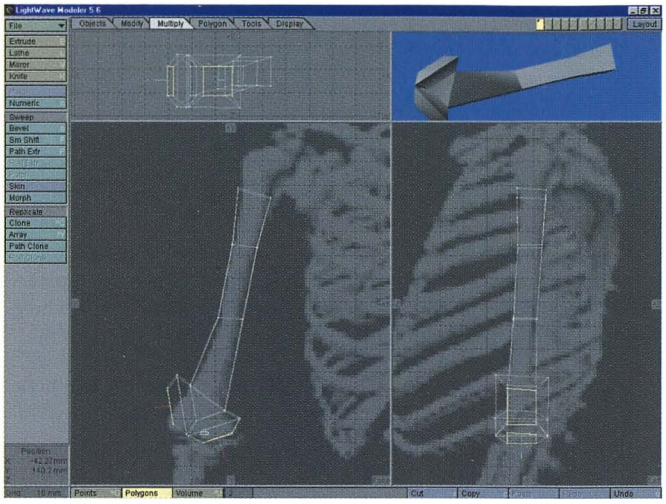


Figure 3: The roughly beveled knobs.

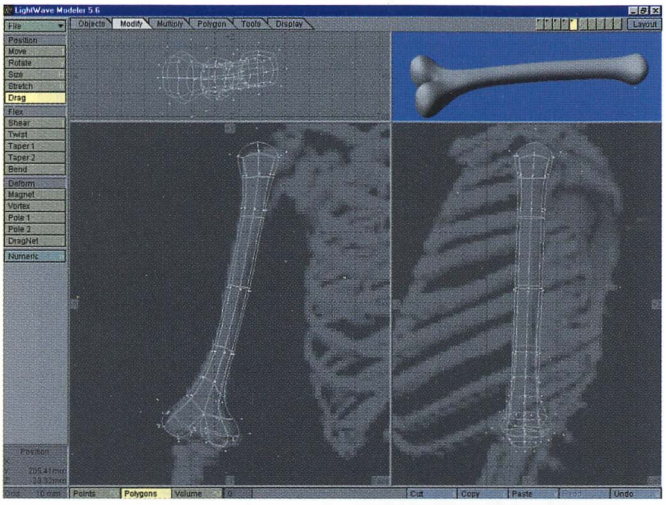


Figure 4: The final metaNURBed Humerus.

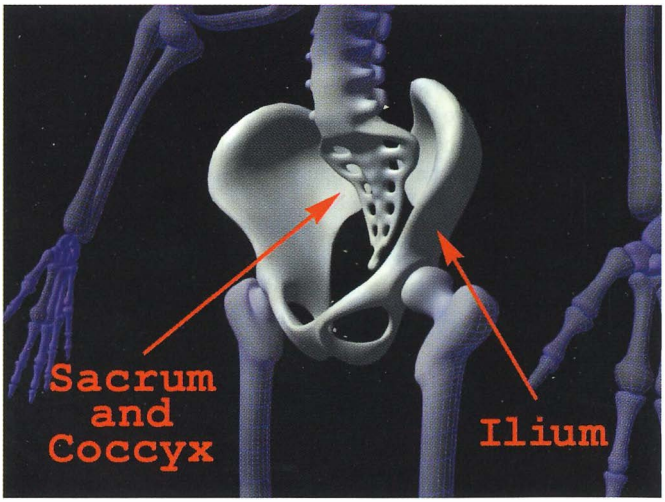


Figure 5: The finished nether-regions.

look at the finished product, Figure 5, to get a sense of the shape we're after.

The tailbone has a lot of holes. Some folks get a knee-jerk Boolean reaction when they hear something has holes in it. Forget it. The Boolean is the bane of the quick modeler, as it requires frequent cleanup work and gives poor control over the resulting geometry. So what's the alternative? First, count the holes. Two across, 6 down (for reasons that will soon become apparent, I'm counting the little divot at the bottom of the tailbone as a hole).

As a rule of thumb, if you have a regular grid of holes in an object, it can be made most simply from a box that is (2*number of holes)+1 in each direction. So, we make a box that is five segments across, and 13 segments down. Now making the holes is merely a matter of selecting a few polygons and cutting them out. The holes in the tailbone are somewhat oval, so select the two vertical rows of points. Counting across, they are rows two and five. Using the stretch tool roughly in the center of the object, hold down the CTRL key to constrain the stretch to one axis. Stretch the X axis, which will effectively move those rows closer to the outer edge of the object. This is one of the keys to mastering the deceptively simple stretch tool. Use it on selected parts of your geometry, and it's just another way to move points around. You should end up with an object that looks like a character from the I-Ching, Figure 6.

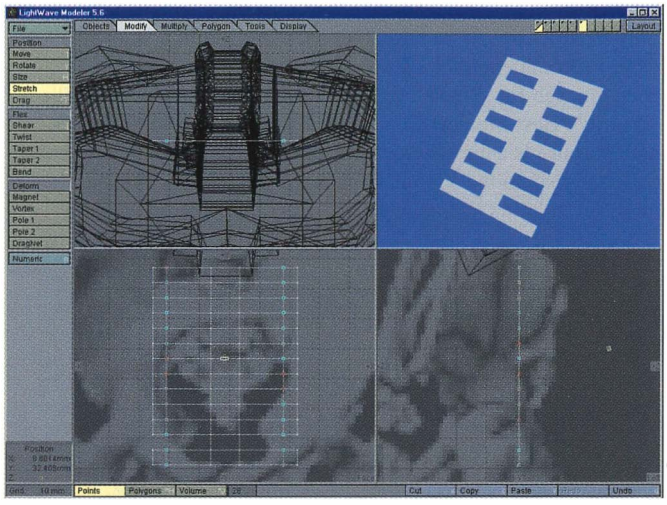


Figure 6: Cleaner than Boolean…make mom proud!

Three tapers will get us into the ballpark, shapewise. And ballpark is the key word here. The reference images I'm using are not lined up with Modeler's origin, which means centering the objects is all done by eyeball (If you feel faint, sit for a minute and catch your breath.). Before the tapers, a little depth is in order. A quick Z+ extrude, and we have an object, not a shape.

Using Volume mode, select the top five rows. Using the Taper 2 tool, taper down the Y axis, with sense -, and the preset curve set to ease-out. As it turns out, I tapered the X value by 53%. Who cares. It looks right, and that's what counts. To set up for the next taper, select the stuff that didn't get tapered last time, and stretch them inward until they line up with the bottom of the previous taper, Figure 7.

Raise the bottom of the selection to avoid the bone at the lower tip of the tailbone, and taper using the same basic Taper 2 settings, until the bottom is nice and narrow. Then select the tip bone, and stretch it down until it seems to fit with the rest of the object. A quick tab test shows a metaNURBed object that looks like some deranged medical spatula.

We're not done yet. There are some areas that are a bit pinched, and the whole thing is still too regular. Before the third taper, drag points around, free up some of the pinched areas, and make the tip bone a little rounder.

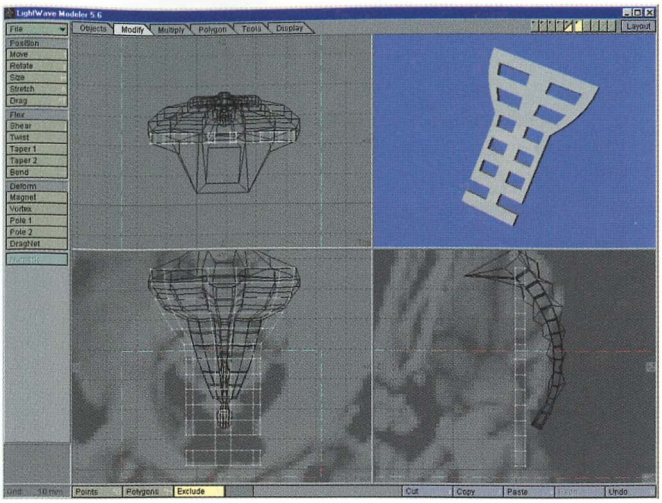


Figure 7: The top has been tapered, the bottom stretched to match.

Here is a good trick to give this piece a nice bit of bumpy, organic depth. Select all the points in the middle two rows. Then, with the top row selected, deselect every other row. Stretch them out a bit on the Z axis. Then, move the points down the Y about half a row's width, to center the peaks of the new bumps over the bony struts. Cheap! Easy! Fast!

Now, one last taper. The bone grows thicker toward the top, and size-wise, it needs to weld onto the spine that I've already created. Fire up Taper 2 again, this time on the whole object. Using the linear preset, widen the bone on the Z axis. From here on out, it's all eyeball work. The outer edges too smooth? Use the same alternate-row stretch trick to make them bumpier. Bones too square? Stretch a pair of rows to round those corners. If all else fails, drag some points. The final act is to bend the tailbone to give it some curvature, Figure 8. Had we modeled the spine in this tutorial, we'd now be doing a little orthopedic attachment surgery, but I'll leave that to your imagination.

Sean Moyer can be contacted through the Electron Dreams website, chromecow.electrondreams.com

Keyframe #17 was missing the following two images from Matt Morgan's F-14 tutorial.

LiftUp

This shot was just way too simple. With the amount of fog, and with the plane being in shadow for most of the shot, I mostly had to worry about the movement and the jet exhaust. I didn't need to bother with any heat distortion, as it was not noticeable from the reference footage. To get the slight highlight on the F-14s exhaust, I used a similar technique for 2D Lighting, as I did with the highlight for the clouds in the TwoSwing shot. I also added an animated color gain in composite to help out the shadowing changes the F-14 goes though as it lifts up.

Figure 7: LiftUp final shot.

RaceCam

I really didn't complete this shot, as I hadn't fully finished my tweaks with the surface object. In my preview renders, the F-14 looked a little like die-cast metal. When I checked my initial composite on video, everything looked fine; so I stopped working on it. It's kind of humorous to me, but when I saw it actually moving in the format it was designed for, those few frames where it looked like die-cast metal just weren't noticeable, Figure 8.

Figure 8: RaceCam final shot.

IFW Shaders

Intel & Alpha
Price : $75
Available only online at www.Shaders.org

reviewed by Lee Stranahan

As an animator, you've probably spent lots of late nights working in front of the computer. You're up until four in the morning, blearily tweaking settings, hitting the F9 key to re-render, all the while with the TV blaring in the background. At that time of night, you get a lot of those commercials from companies selling Ginsu knives, cassette tapes telling you how to get rich, and collections of every hit song ever written.

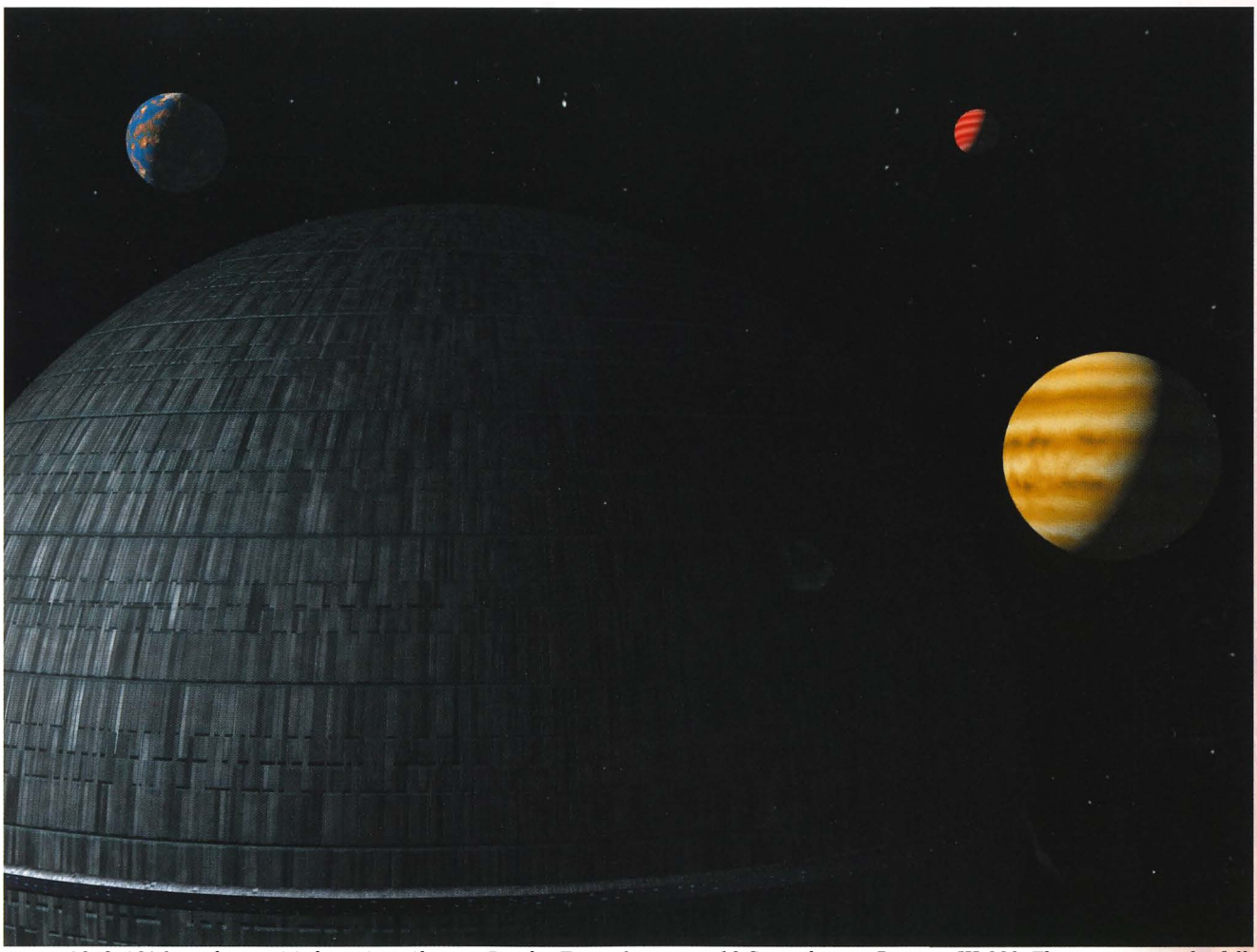
I was reminded of these commercials as I wrote this review for the IFW Shaders, because this is a plug-in set that gives you so much bang for the buck, that just trying to describe it factually sounds like the overblown script from a Ron Popeil infomercial. So, I'll just give in and deliver the review like a late night sales pitch.

Gather round, texture lovers. You say you want to create exciting new surfaces, but are bored with only a couple of dozen procedural textures? Do you like the realism of image maps, but don't like the memory they use? Or are you the kind of person who loves plug-ins the way kids love Christmas presents, but who doesn't want to spend a ton on your toys?

Well, folks, have I got a plug-in set for you: IFW Shaders. Over 120 new procedural textures, perfect for every occasion. Now, some of you may be asking exactly what a procedural texture is. Basically, it's a texture created with a mathematical formula. LightWave includes dozens of 'em, from simple ones like Checkerboard to useful ones like Fractal Noise.

The IFW Shaders expand your palette by giving you dozens and dozens of new texture possibilities. You add them as 11 different plug-ins, and then access them in LightWave in the Shaders section of the Surface panel.

Continued on page 28



IFW_Space *2048x1536 resolution, Medium Anti-Aliasing Render Time: 6 minutes, 18 Seconds on a Pentium III 550. This image uses the following FW Shaders: no image maps are used, IFWStarField (Stars in background), IFWPlanet (Earth-like planet), IFWGasGiant (Red Planet & Orange Planet), IFWDeathStar (The Big Metal Thing), IFWBoilerPlate (The Big Metal Thing), IFWHull (The Big Metal Thing), IFWWindows (row of windows on The Big Metal Thing).*

FOR DIGITAL VIDEO, ANIMATION, MULTIMEDIA AND WEB DEVELOPMENT

A Safe Harbor Exclusive!

Lip Service – A unique software package that enables animators to quickly and easily generate a multitude of facial expressions, mouth movements, and phonemes, all with realistic results! *Includes Shave And A Haircut* **286.00**

Shave And A Haircut – Features include Shadowing, Motion Blur, Collision Detection and more.**285.00**

Vue d' Esprit 3 – This powerful 3D application is optimized for creating, rendering and animating natural scenery. **Vue d' Esprit 3** was designed for the beginner, as well as the advanced user. It is fast, easy to use, and includes many new features, such as animation, elaborate timeline, and professional video options. Also, fully textured objects created with LightWave 3D can now be imported and exported easily **189.00**

A Safe Harbor Exclusive!

Oxygen GVX1- AGP
A 100% geometry and lighting acceleration card which integrates the GLINT R3, and the GLINT Gamma G1 processor on a single AGP board **699.00**

Oxygen GVX1- PCI.....

Oxygen GVX210 AGP – Delivers high-end geometry and rasterization graphics acceleration on a single AGP board **1349.00**

Oxygen VX1 AGP – Achieve 2D and 3D superb graphics with the GLINT R3 processor **209.00**

Oxygen VX1 PCI ....

Canopus DVRex-M1/Premiere - Combines high-quality DV video with multi-channel audio support; includes Premiere 5.1, Kick Start Training CD, breakout box, Rex Edit 2.5, and cables.....**2399.00**

Canopus Rex RT Option - Add real-time effects to your DVRex editing system!.....**999.00**

EZDV **249.00**

Rocket Rex **99.00**

Mega Production Bundle - All-in-one super package includes DVRex-M1 breakout box, Rex RT option, Rex Edit 2.7, Adobe Premiere 5.1, Rex Navigator software, and all cables.....**2999.99**

LIGHTWAVE PLUG-IN/ LIGHTWAVE RELATED SOFTWARE

3D Super Bundle (Art Recon)	529.00
AuraFX Vol. 4	99.99
Alpha Magic Wizard	199.00
Amazing Sci-Fi Textures	103.99
Animation Master 2000	269.99
Animation Master/Simbiont Bundle	319.00
BlowUp & Shatter For LightWave	89.99
Boris FX 5.0	399.00
Commotion 3.0	1995.00
Complete Effects Library	369.00
Creature Creator for LightWave	225.00
DarkTree Textures	289.00
Dinosaurs ROM CD PLUS!	60.00
FiberFactory3	269.00
Genesis V2 Pro	269.99
Graffiti	399.00
[illegible]	Call
	Call
Life Forms Studio 3.9	399.99
Light ROM Super Bundles	Call
MacroForm 2.0, Intel	289.00
ReelMotion 1.0	634.99
	699.00
Particle Storm 2.0	489.00
Power CG Plus/Speed Razor	295.00
Project: messiah	549.00
Relativity	189.99
Seamless Textures 6 – Architectural Ornament	129.00
SurfacesROM	34.00
Tree Druid, Intel/Mac	169.00
Visual Effects CD Set	132.99
World Construction Set 5.0 Intel/Dec	

THE JOY OF SIX!

LightWave [6]

The new version of the award-winning 3D animation system from NewTek! Powerful new tools include enhanced interactivity and scene editor, Skelegons, MetaNURBS in layout with point weighting, caustics, UV Mapping, surface editing in modeler and a whole lot more! (specifications subject to change without notice) LightWave 3D is the most-used animation system on the planet! What are **you** using? **2250.00**

LightWave [6] Upgrades .....**Call**
Aura 2.0 .....**895.00**
Aura 1.0 - 2.0 Upgrade **295.00**
Aura Powerhouse Bundle – Includes Aura and AuraFX combo set! **799.00**
Calibar .....**Call**
Inspire 3D/LD_Tools Bundle **459.99**

Visit our website
www.videotoasternt.com

Video Toaster NT

Never again compromise the quality of your video - lossless video editing is now available on your desktop! Introducing Video Toaster NT - the next generation, uncompressed digital video solution from NewTek. For only $2995, your complete post-production studio includes:
• Speed Razor (award-winning non-linear editing) • Aura (2D graphics, manipulation & compositing) • Lightwave (professional 3D animation)
Already own a Video Toaster, Flyer or Lightwave? Save $500 when you upgrade to the Video Toaster NT!
Complete reliable, quality turnkey Toaster workstations are also available!

Video Toaster NT SDI Option **995.00**

Pre-order NOW!

Video Toaster NT 1.0-2.0 .....**495.00**
Video Toaster NT 1.0-2.0 + Breakout box **1995.00**

Cinematte for Aura

New plug-in for Aura! The acclaimed blue screening tool that allows you to composite images together while maintaining proper anti-aliasing and transparency is now available!. Works with Aura 1.0 and 2.0.....**239.00**

dpsReality

The world's first Studio Digital Disk Recorder! Features compressed or uncompressed recording, alpha channel, balanced/unbalanced stereo audio I/O and DPS Virtual Tape File System for group rendering. The package contains the dpsReality player application, Digital Fusion DFX compositing software, and an audio/video breakout cable. **2199.00**

Reality™ Studio Digital Disk Recorder

DPS™

dpsVelocity – 2D

The combination of hardware and software is what makes this real-time NLE system the ultimate choice. The **dpsVelocity – 2D** delivers faster, sharper, and easier edits at an affordable price. This real-time editing solution includes the dpsReality card, dpsReality player, audio/video breakout box cable, dpsVelocity NLE software, and so much more **2699.00**

Velocity™ FASTER, SHARPER, EASIER EDITS

dpsVelocity – 3D **4499.00**
dpsVelocity NLE Software.....**805.00**

Visit our site
www.dvguide.sharbor.com/dps/
for more information.

Terms: POs accepted from schools and government agencies • No waiting for personal or business checks • Defective products replaced promptly. RMA number required (call 262-548-8159) for all merchandise returns. Returns not accepted after 15 days. Returned products must be in original packaging, postage prepaid. Opened software not returnable. Shipping charges not refundable. Returns subject to a 15% restocking fee.• Not responsible for typos. Prices subject to change.

Information: 262-548-8120
Fax 262-548-8130
Tech Support / RMAs:
262-548-8159, 12-4 pm Mon.-Fri.

Safe Harbor Computers
W226 N900 Eastmound Dr.,
Waukesha, WI 53186

IFW Cows *2048x1536, Low Anti-Aliasing Render Time : 4 minutes, 37 seconds on a Pentium III 550. This image uses the following IFW Shaders: no image maps are used, IFWDinoSkin (Cowasaurus), IFWScales (Cowsaurus), IFWCellular (Cow With Extremely Bad Acne), IFWMonster (Disgusting Bloody Cow).*

They work with Intel and Alpha versions of LightWave 5.6, 6.0 and even NewTek's 'baby LightWave' program, Inspire.

So, what kind of procedurals do you get? Everything from stone walls to stucco to stained glass to stars! There are scads of trippy psychedelic textures to impress your friends who attend raves or SIGGRAPH meetings, including Mandelbrot, Julia and Newton sets. There are textures to simulate crushed velvet, leather, or plaid. There are bathroom tiles, old bricks, brushed metals, dirt, and more. There's a lot of variety, and every texture has literally millions of variables. On top of that, these shaders can control all of LightWave's surface parameters (diffusion, specularity, etc.) and can be mixed, matched and combined in all sorts of ways. You have more texture possibilities than you can shake a forest at. But the bottom line is still, "How do they look?"

The answer is, "Very, very good." Procedurals have some built-in limitations, but the IFW Shaders hold up pretty well. Procedurals are generally best used as part of an overall surfacing strategy, not as the only surfacing technique. Used this way, the IFW Shaders look fantastic and can really add another layer of realism. There are a couple of textures that I wasn't able to tweak into a useful result, but most of the textures will have a use one way or another in your work.

But don't order yet...

For you sci-fi maniacs, IFW Shaders also include the Space Series. There are shaders to create starfields, planets of every description, laser blasts, and more. There are also a handful of textures that are ideal for creating the kind of starship textures that artists normally use Photoshop to create.

Imagine if Ron Thornton's brain was digitized and turned into an algorithm. Then you'd get a collection of procedurals like IFWNurnies, IFWHull, IFWBoilerPlate, and IFWDeathStar. These enable you to create those cool panels that seem to cover every LightWave-rendered spacecraft since Ron launched *Babylon 5* into orbit. These textures also work together.

But wait...there's more!

You want to be able to animate your textures? The IFW Shaders give you more envelopes than your local post office. Pretty much every parameter that you'd want to animate, you can animate using standard LightWave plug-in envelope controls. Further, there are a number of shaders that were born to animate; these are cleverly called the IFW Animated Series. The Animated Series all have special parameters that create some very useful effects. Using these shaders, you can create walls of fire, banks of bouncing LEDs, the effect of someone 'beaming up' to a spaceship, or rain drops.

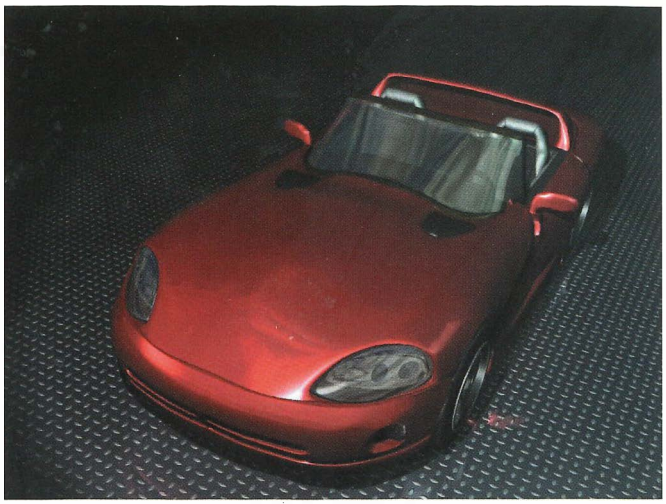
Now what would you expect to pay?

The price is one of the best features of the IFW Shaders, but this collection is not available in any store! The only place to buy the shaders is to direct your web browser to

www.Shaders.org and order your copy there. For $75, you not only get the whole collection of shaders, but you also get free updates for six months, plus access to tips and tech support by email. The free updates aren't just an idle feature – there really are new shaders being released all the time. The tech support has been great, too. I've asked a couple of questions and have always gotten a response within 24 hours.

The only downside to this kind of personal technical support is that if these shaders start to garner the popularity they deserve, the number of questions will increase exponentially. To that end, I've personally started a free mailing list for users. The idea is to divert some of the tips and tricks traffic to other users. If you're interested, sign up at: **http://www.egroups.com/subscribe/IFWShaders**

I can wholeheartedly and without reservation recommend this package for anyone who loves 3D surfacing. This review only hints at the power of this package. Go to Shaders.org and check out the impressive gallery of IFW-inspired images, as well as the complete index of all the shaders. This package gives you a fleet of new tools to play with, all fully animatable, with great support at a killer price. This is exactly the kind of development that should be supported, where the developer and user can have a close working relationship. It's a great product and a lot of fun.



Viper *This image uses the following IFW Shaders in addition to a variety of surfacing techniques, including Worley Lab's* ***Gaffer*** *– it's included to show a more contextual use of the IFW Shaders. Model by MeshFactory.com, IFWStamped (The Floor Under The Car), IFWLeather (Upholstery – hard to see, but I swear it's there.)*

CINEMATTE

Digital Dominion
1910 Byrd Avenue, Suite 214
Richmond VA 23230
dgdominion@aol.com
http://users.aol.com/dgdominion
800-462-4369
804-282-1157
$249.95

reviewed by Doug Kelly

Cinematte is a bluescreen compositing plug-in. It has been available for Photoshop for a while, and the port to Aura retains most of the Photoshop structure, including Cinematte's installation as an External Filter.

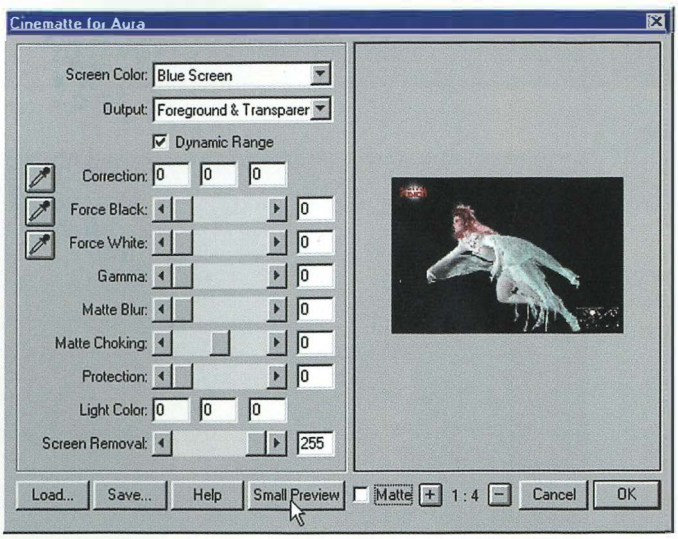


Figure 1: Cinematte external filter plug-in interface.

The documentation is sparse but adequate for the experienced Aura user. Other than an instruction to use the Setup executable on the single installation diskette, you are left to figure out installation on your own. If you have never installed a Photoshop plug-in for Aura, you will need to work through the Aura manual section on the subject (page 8.17 in the 1.0 manual). However, if you know what you are doing, the installation is fast, painless and occupies only a few hundred K of your hard drive. Let's hear it for lean code!

The Cinematte manual is definitely for the experienced compositor, not for the novice. The once-over-lightly in this manual is just fine for busy professionals; it will have you up and running in a matter of minutes. All the terms of art mean exactly what you expect them to, happily avoiding marketing neologisms in favor of standard terms. The manual starts out with a simple features-in-order-of-controls. You can learn the basic operations by loading some bluescreen footage into Aura, selecting the Cinematte external filter, and reading through the first nine pages of the manual while tweaking controls.

If you don't know what a *matte* is, or are still fuzzy on the precise meaning of the term *gamma*, you'll need to beef up your working vocabulary. I recommend Ron Brinkmann's *Art and Science of Digital Compositing* for the theory, and my own *Digital Compositing In Depth* for a hands-on series of tutorials that you can repeat with Cinematte. By the way, I'm very happy to see Cinematte available for Aura. Aura 1.0 did not fare well in comparison to many other compositing programs in my book, partly due to Aura's lack of native bluescreen tools. With Cinematte installed, Aura is a much more useful compositing tool.

I have a few small quibbles with the way the Cinematte interface works, Figure 1. As with many RGB-based graphics programs, the sliders are designed for a range of 0 to 255. However, the Cinematte documentation clearly states that many of the parameters have much smaller useful ranges. I think it would be more intuitive if the sliders were designed for the normal useful range for each parameter, and leave the theoretically useful maximum of 255 accessible only through the data entry fields. This would make the sliders more useful and friendly to beginners and those in a hurry, while enabling the more technically minded to play with extreme numbers to their heart's content.

I also find it annoying that the preview window does not support magnification larger than 1:1. I believe it is absolutely essential that a compositor be able to zoom in at least 4X to examine transition areas. If you can't see fine details in the preview, you are reduced to repetitive, time-wasting trial and error. I hope this oversight is corrected in the next revision of Cinematte; judging by the screenshots in the documentation, a more useful range of preview zooms is already

available in the Photoshop version of this plug-in.

After about half an hour of tweaking controls and reading through the manual, I was able to get acceptable results from some demo footage, Figure 2. This comp used the settings shown in Figure 3.

Cinematte seems to have the features necessary to successfully pull a matte from blue- or green-screen footage (I didn't have any test footage for the more rare orange-screen setup.). While the controls for eroding, choking, blurring or changing the gamma of the matte are touchy, they do their respective jobs. The Light Color feature does a very nice job of spill removal without wiping out high-end colors.

Overall, I found Cinematte to be a solid tool for bluescreen compositing. All the required features are present and work well, and the learning curve is short enough that busy professionals can add it to their toolbox for a modest investment in time. The price is reasonable for the earning power it adds to Aura. If you are an Aura user and need to composite bluescreen footage, you'll get your money's worth from Cinematte.

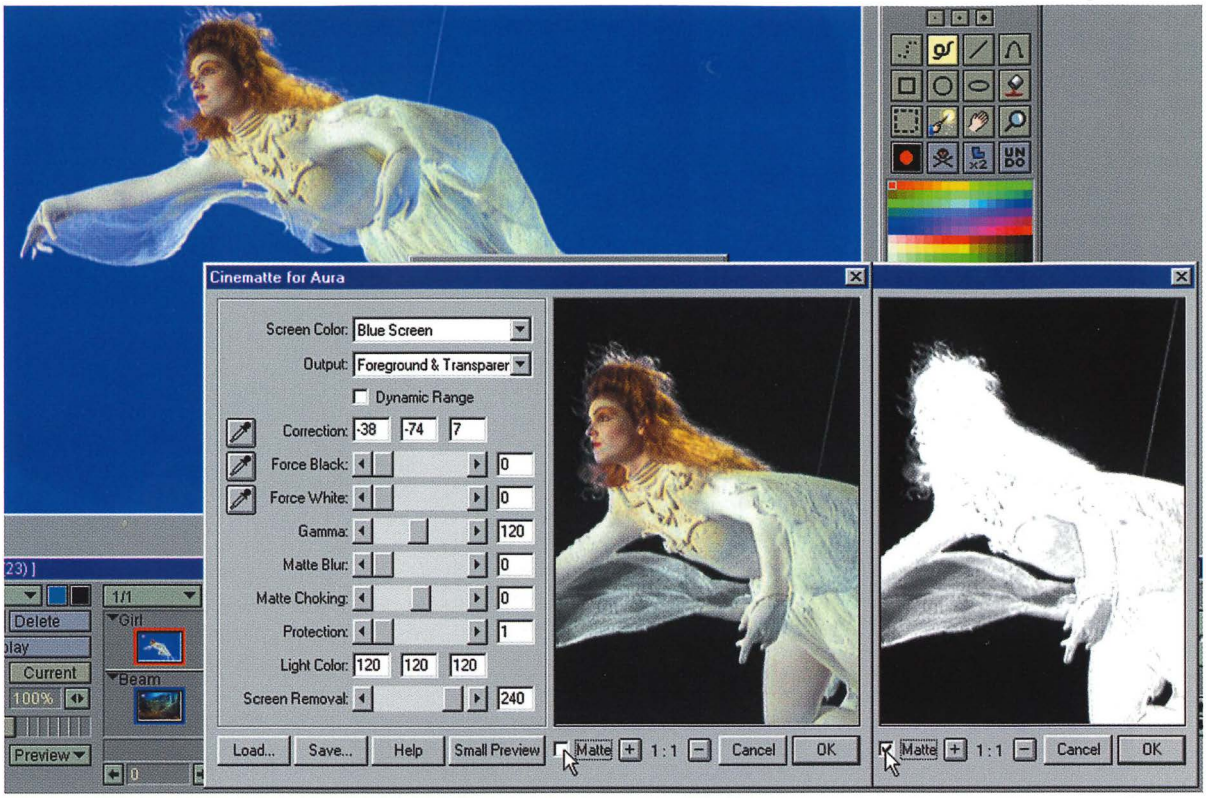


Figure 3: Settings for test comp. Inset right shows matte view.

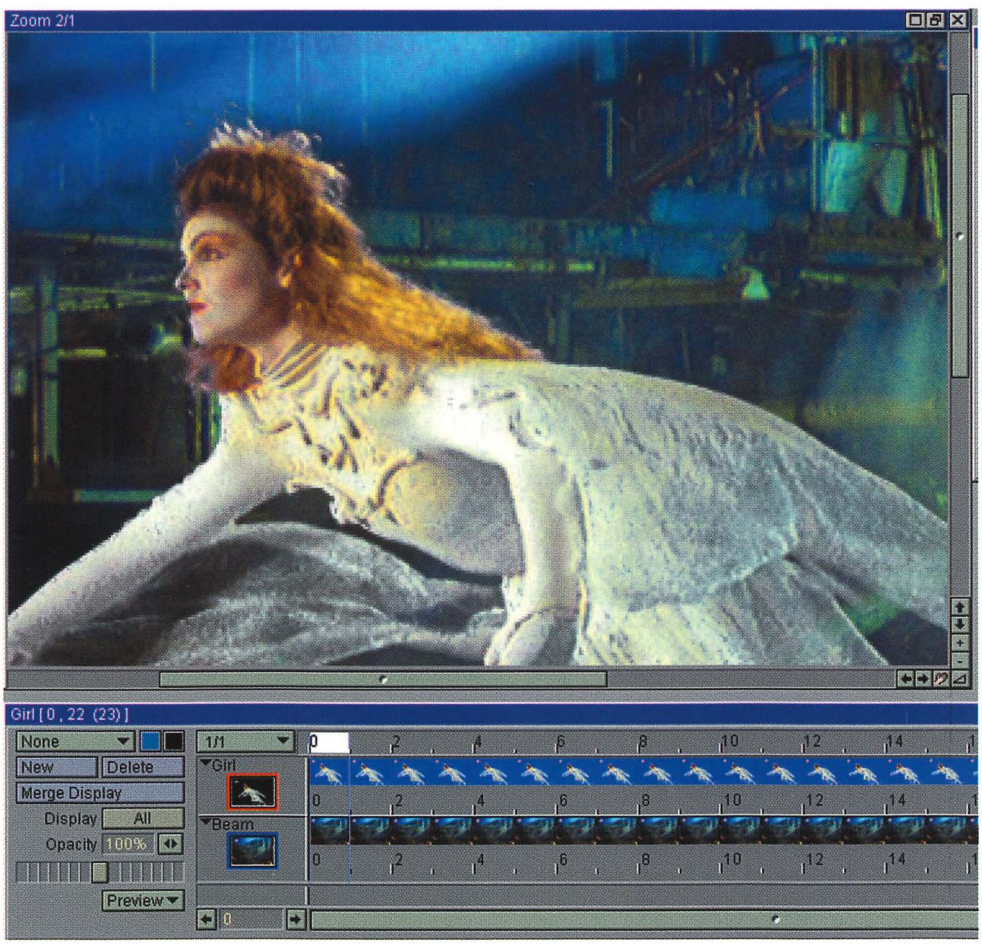


Figure 2: Test comp of bluescreen footage over a complex background.

Continued from Page 7

The LightWave Mailing List Search Engine CD contains nearly 87,000 messages compiled in a format that is easy to search for any topic that has been discussed in the past 3 years. It is available now for only $19.95 and is a valuable information resource for all LightWave users. You can find out more by visiting www.tv3d.com.

For more info about David Warner or TutorVision 3-D CD, his series of multimedia tutorial CD-ROMs for LightWave, and a FREE TutorVision Sampler CD, please visit http://www.tv3d.com or send some e-mail directly to Dave at dave@tv3d.com

Celebrate L6 With Us!

One FREE Issue to All New Subscribers and Renewals!

The Men Behind the Curtains

by Steve Pugh

Now that the dust has settled some from NewTek's release of LightWave 6, the newsgroups are again aflurry with kudos, gripes, and outright battles over the program's features, fixes, upsides and downsides. From some of the posts, it's easy to tell that sometimes folks forget one of the most amazing things about graphics software: It all comes out of someone's brain.

Instead of waxing philosophical about the effects business, I think I'll look at the guys behind all that code. I direct your attention to Exhibit A, the big, gray lumps of wet neurons that inhabit the skulls of these gifted byte-wranglers. I've done a little bit of programming here and there, mostly utility stuff: that beige, functional code that moves files, reads databases, and writes log entries. But every now and then I'll stop and think about what the hell has to go on inside your head when you sit down to write a fur shader, or a radiosity engine.

Sometimes it takes me back to grade school and the wide-eyed amazement that accompanied so many new things - the first time I saw someone flip their eyelids inside out, the first time I heard someone talk in Pig Latin. Think about it; somebody actually had to think that stuff up! Somebody, somewhere sat down with a pad of paper (probably more than one), and came up with the idea that a bunch of variables and formulae could make a picture of a chrome teapot on a checkerboard plane.

Every time you feel particularly pleased with a scene you've animated with Hypervoxels, think for a second about what kind of recipe was needed to write that volumetric renderer. "OK, here's the origin of the Null object and its Size and Rotation. Based on the distance to the camera, and the frame number, and these texture size and animated turbulence settings, and the speed of that Null over there, this dot should be a little more green than it was a frame ago. Never mind the fact that nobody's added any actual geometry to the scene."

I know that right about now Gregory and Allen from NewTek (and a whole host of others) are chuckling at the staggering inaccuracies in that savaged interpretation, but I hope they'll cut me some slack. My point isn't the exact formula, it's the fact that someone had to come up with that stuff practically from scratch. Even the formulas that are out there for all to examine and reuse had to be invented at some point.

Some guy woke up one morning and thought "Hey, it'd be really cool to point a bunch of cameras at reflective balls on some dude's arms. You could compare where each ball was, from a bunch of different angles, and figure out where it was in 3D space based on where the picture was taken!" A bunch of pens and paper and coffee later, optical motion capture is a reality.

So give pause the next time you're watching *A Bug's Life* or *Episode One*, and while you're marveling at the artists behind those characters and environments, take a second to tip your hat to the artists behind those artists. They deserve it.

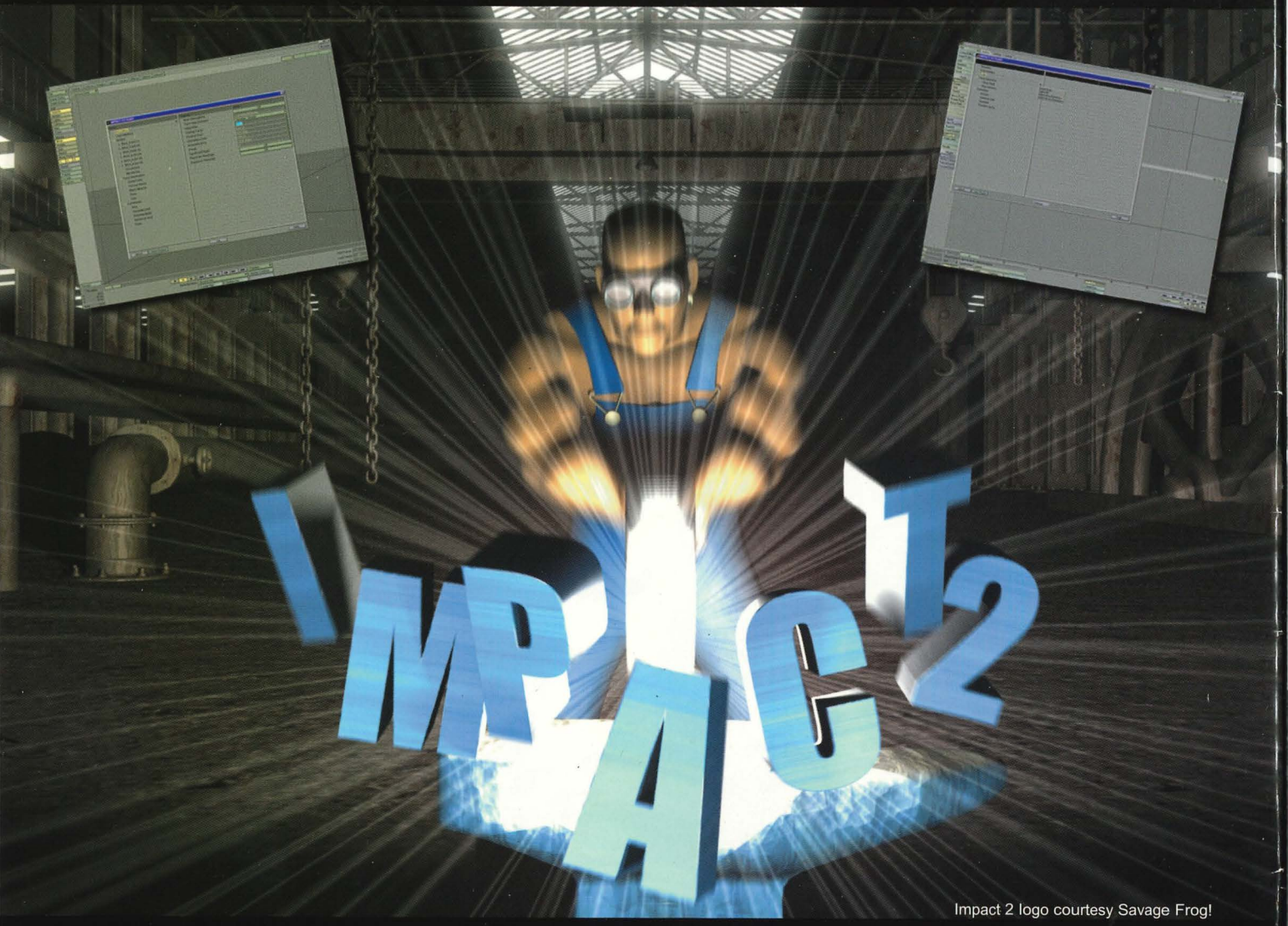
Steve Pugh is the General Manager at Foundation Imaging. If he's not daydreaming or fighting a mental block, you can contact him through their website, www.foundation-i.com.

- Uncompressed D1
- SpeedRazor™ SE Non-linear Editor
- Aura™ Video Paint
- LightWave® VT

www.videotoaster.com
Call **(800) Toaster**
Outside the U.S. call (210) 370-8000

NewTek

Rigid Body Dynamics for LightWave 3D 5.6 and 6.0!



Impact 2 logo courtesy Savage Frog!

Breaking the barriers of rigid-body dynamics!

Completely redesigned and integrated in LightWave 5.6 and 6.0.

-Advanced collision detection and response-
-Several constraints such as spring; rods, pin joints, and more...
-Varying force generators-

Complete feature list and pricing at www.dynamic-realities.com

Dynamic Realities

Complete feature list and pricing at

www.dynamic-realities.com